

**Expanding Software Process Improvement Models
Beyond the Software Process Itself**

A DISSERTATION PRESENTED TO

THE DEPARTMENT OF INFORMATION SYSTEMS
UNIVERSITY OF CAPE TOWN

IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE

MASTER OF COMMERCE DEGREE
IN
INFORMATION SYSTEMS

by

C. SAKS

October 1998

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

This dissertation is not confidential and is non-propriety.

I certify that the dissertation is my own work and all references are accurately reported.

C. Saks

Synopsis

The problems besetting software development and maintenance are well recorded and numerous strategies have been adopted over the years to overcome the so-called "software crisis". One increasingly popular strategy focuses on managing the processes by which software is built, maintained and managed. As such, many software organisations see software process improvement initiatives as an important strategy to help them improve their software development and maintenance performance. Two of the more popular software process improvement (SPI) models used by the software industry to help them in this endeavour are the Capability Maturity Model for Software (SW-CMM) from the Software Engineering Institute and the Software Process Improvement and Capability dEtermination (SPICE) model from the International Standards Organisation.

This research begins with the supposition that, although these SPI models have added significant value to many organisations, they have a potential shortcoming in that they tend to focus almost exclusively on the software process itself and seem to neglect other organisational aspects that could contribute to improved software development and maintenance performance. This research is concerned with exploring this potential shortcoming and identifying complementary improvement areas that the SW-CMM and SPICE models fail to address adequately.

A theoretical framework for extending the SW-CMM and SPICE models is proposed. Thereafter complementary improvement areas are identified and integrated with the SW-CMM and SPICE models to develop an Extended SPI Model. This Extended SPI Model adopts a systemic view of software process and IS organisational improvement by addressing a wide range of complementary improvement considerations.

A case study of an SPI project is described, with the specific objective of testing and refining the Extended SPI Model. The results seem to indicate that the framework and Extended SPI Model are largely valid, although a few changes were made in light of the findings of the case study. Finally, the implications of the research for both theory and practice are discussed.

Contents

LIST OF FIGURES iv

LIST OF TABLES v

1. THEORY 1

 1.1 Introduction 1

 1.2 Background 1

 1.2.1 The Growing Interest in SPI 2

 1.2.2 Software Process Improvement Models 3

 1.2.3 The Impact of SPI 5

 1.3 Perceived Shortcomings of the SW-CMM and SPICE Models 7

 1.4 Proposed Extensions to the SW-CMM and SPICE Models 13

 1.5 The Purpose of the Research 20

2. LITERATURE SURVEY 24

 2.1 Overall Structure of the Proposed Framework 24

 2.2 Change Environment 27

 2.2.1 Strategic Initiatives 27

 2.2.2 Change Resources 30

 2.2.3 Change Facilitators 31

 2.2.3.1 Learning Capacity 32

 2.2.3.2 Cultural Readiness 33

 2.2.3.3 IT Leveragability and Knowledge-Sharing Capability 34

 2.2.3.4 Overcoming Knowledge Barriers 35

 2.2.3.5 Positive Management Orientation 37

 2.2.4 Change Context 39

 2.3 Change Implementation 44

 2.3.1 Software Process Dimension 44

 2.3.1.1 Software Process Improvement 44

 2.3.1.2 The SW-CMM and SPICE Models 49

 2.3.2 Non-Process Change Dimensions 59

 2.3.2.1 Structure 60

 2.3.2.2 People 67

 2.3.2.3 Information and Technology 70

 2.3.2.4 Management 75

 2.3.2.5 Other Enablers 78

 2.3.3 Change Management 82

 2.4 Change Outcomes 88

2.5 Literature Study Summary 89

3. METHODOLOGY 94

3.1 Introduction 94

3.2 Site Selection 95

3.3 Methodology Procedures 96

4. FINDINGS 104

4.1 Background 104

4.2 Findings 108

4.2.1 Summary of the Findings 108

4.2.2 Details of Findings 114

4.2.2.1 Change Environment 114

4.2.2.2 Change Implementation 140

4.2.2.3 Change Outcomes 164

4.3 The Updated Extended SPI Model (ESPIM) 167

5. CONCLUSION 171

5.1 Implications and Limitations of the Research 172

5.1.1 Implications for Theory and Research 172

5.1.2 Implications for Practitioners 176

5.1.3 Limitations 178

5.2 Concluding Remarks 180

REFERENCES 182

APPENDIX A: INTERVIEW SCHEDULE 189

List of Figures

Figure 1-1: Business Process Change Model from Kettinger and Grover (1995)	8
Figure 1-2: A Proposed Software Process Improvement Framework (SPIF)	15
Figure 2-1: Organisational change process for the adoption of CASE tools (Orlikowski, 1995)	25
Figure 2-2: SPICE's SPI Steps (ISO, 1996g)	47
Figure 2-3: The IDEAL SPI approach (Gremba & Myers, 1997)	48
Figure 2-4: The main elements of the SW-CMM and SPICE models (adapted from Anonymous, 1997b)	49
Figure 2-5: Mapping the SW-CMM maturity levels to SPICE capability levels	58
Figure 2-6: Process Improvement Team Relationships (adapted from Orville, 1998 and ISO, 1996g)	64
Figure 2-7: Model of Learning Strategies (Baldwin et al, 1997)	69
Figure 2-8: Two learning loops for process information (Davenport & Beers, 1995)	76
Figure 2-9: Eight steps to organisational transformation (Kotter, 1995)	85
Figure 4-1: The scope of the SPI project described by the case study	105

List of Tables

Table 1-1: Imperatives for the IT organisation from Rockart et al (1996)	10
Table 2-1: Proposed learning approaches in the context of SPI	33
Table 2-2: Top Managers' Orientations (Choi & Behling, 1997)	37
Table 2-3: Categorisation of the Institutional Context (Orlikowski, 1993)	40
Table 2-4: Generic Maturity Level Definitions (adapted from SEI, 1995 and ISO, 1996b)	51
Table 2-5: Key process areas of the SW-CMM by maturity level	52
Table 2-6: SPICE process categories and processes	53
Table 2-7: SPICE capability levels, process attributes and management practices	55
Table 2-8: The advantages and disadvantages of the SW-CMM and SPICE architectures (adapted from Paulk et al, 1995 & 1996)	57
Table 2-9: The Extended SPI Model	90
Table 2-10: Mappings from the software process maturity or capability levels to the non process change dimensions	92
Table 3-1: Case study participants	101
Table 3-2: Documentary evidence sought during the case study	103
Table 4-1: Summary of the case study observations	109
Table 4-2: Areas for which insufficient evidence was gathered to meaningfully comment	114
Table 4-3: The Updated Extended SPI Model	168
Table 4-4: Updated mappings from the software process maturity or capability levels to the non process change dimensions	170
Table 5-1: Research objectives	181

1. Theory

1.1 Introduction

The disciplines incorporated in software process improvement (SPI) seem to offer much promise in addressing the so-called "software crisis". Although SPI is a relatively recent approach to improving the performance of software organisations, indications are that organisations are already deriving significant benefits from their SPI efforts.

Software process improvement models, such as the Capability Maturity Model for Software (SW-CMM) and the Software Process Improvement and Capability dEtermination (SPICE) model, are increasingly being used by software organisations to guide and assess their software development and maintenance performance improvement efforts. These models, however, appear to have a very narrow focus on the software process itself and do not seem to address issues pertaining to the environment within which SPI occurs or other areas that could contribute to improved software development and maintenance performance. This research explores this perceived gap, seeking to identify areas in addition to the software process that contribute to improving the performance of software organisations. It specifically explores the relationship between these areas and the SW-CMM and SPICE SPI models.

1.2 Background

This section describes the emergence of software process improvement as a response to problems in the software community, examples of models that have been developed to guide SPI exercises and some findings on the impact of SPI on organisations.

1.2.1 The Growing Interest in SPI

Providers of software have consistently been criticised over the past two or three decades for their inability to satisfy the demand for high quality software products in a timely and cost efficient manner (Boggis & Yarrington, 1997). This persistent "software crisis" has become more critical as software becomes more pervasive.

The challenges facing software developers are overwhelming. The demand for ever more complex and sophisticated software is accelerating (Boggis & Yarrington, 1997), technology is advancing at a breath-taking pace, and the rapidly changing software-hungry business environment is becoming increasingly global, competitive and consumer oriented. As a result the "ability to develop and deliver reliable, usable software within budget and schedule commitments continues to elude most software organisations" (SEI, 1995). It is therefore not surprising that improving the effectiveness of software development has consistently been identified as a key issue in IS management by the Society for Information Management (SIM) surveys since 1980 (Brancheau, Janz & Wetherbe, 1996).

Despite new and improved software technologies and methodologies the problems of software development and maintenance have persisted. Many organisations now realise that their fundamental problem might not be about technology or methodology *per se*, but rather an inability to manage the software process (SEI, 1995). This has lead to a growing interest in applying process management principles to the software process and, in particular, an interest in SPI over the past 10 years.

Indications of this growth of interest by the software community to adopt SPI as a means to improve software development and maintenance performance include:

- The growing number of organisations using SPI models such as the SW-CMM to guide their software development and maintenance improvement efforts (SEI, 1997).
- The rapidly expanding interest in organisational units, such as software engineering process groups, that are responsible for SPI initiatives (McGuire & Gibson, 1996).
- Government sponsored SPI programmes such as the European Community's European Software and Systems Initiative (ESSI) (Downes, George & Chappell, 1996).
- The development of International Standards Organisation (ISO) standards focused on software (ISO, 1996a-i).
- The inclusion of a Software Process Improvement track in recent years in mainstream IS conferences such as the Americans Conference on Information Systems (ACIS) (Nerur & Raghupathi, 1997).

1.2.2 Software Process Improvement Models

A number of models have been developed to guide SPI efforts. Authors such as Humphrey (1995), McGuire & Gibson (1997), Walker & Knight (1997) and Anonymous (1997) list some of the better known SPI models as:

- The Carnegie Mellon University Software Engineering Institute's *Capability Maturity Model for Software (SW-CMM)* which is dominant in the USA and, universally, one of the most widely used and influential SPI models.
- The *Personal Software Process*, also from the Software Engineering Institute, that focuses on SPI at an individual or small team level.
- Derivatives of the SW-CMM model such as *Trillium* which was specifically tailored for the telecommunications industry.

- The *ISO9000* series of quality management standards, particularly those aspects that have application to software development (ISO9001 & ISO9000-3). The ISO9000 series is described at the policy statement level and is supported by additional International Standards Organisation and International Electrotechnical Commission (ISO/IEC) standards at a detailed, implementation level. ISO12207, for example, describes software engineering life cycles and their tailoring and improvement.
- *Bootstrap* from Europe which is heavily influenced by the SW-CMM, ISO9000 and the European Space Agency software process standards.
- The *Software Process Improvement and Capability dEtermination (SPICE)* standard which is being developed under the auspices of the International Standards Organisation and the International Electrotechnical Commission in an attempt to describe an international set of standards for process improvement and assessments. SPICE has sought to borrow from the best of all the available SPI models.

In addition to these models a number of individual organisations have suggested SPI approaches of their own. For example, the Bose Corporation (Harkness, Kettinger & Segars, 1996) and the Boeing Space Transportation Systems division (Yamamura & Wigle, 1997) followed process improvement and innovation initiatives focused on quality management, process improvement and innovation and organisational learning. The NASA Software Engineering Laboratory's (SEL) SPI approach is based on the Quality Improvement Paradigm (Basili & Green, 1994; Pajerski & Waligora, 1997; Pfleeger & Rombach, 1994).

Most of these models describe the key characteristics of a core set of software processes and a series of process management practices that should be applied over time to improve the software processes to maturity. A mature software process is one that is repeatable, optimised for the environment in which it is applied and able to be continuously improved (SEI, 1995).

The Capability Maturity Model for Software (SW-CMM) and the Software Process Improvement and Capability dEtermination (SPICE) model appear to be the two best known and most widely used SPI models and were therefore emphasised in this research. In particular the current versions of theses models, namely Version 1.1 of the SW-CMM and Version 2.0 of the SPICE model, were referenced in the research.

1.2.3 The Impact of SPI

The maturity of an organisation's software processes is a predictor its software projects' ability to meet their goals. Projects in organisations with immature software processes experience wide variations in achieving cost, schedule, functionality and quality targets. However, as an organisation's software processes mature under the influence of SPI, improvements in process predictability, control and effectiveness can be expected (SEI, 1995).

- In terms of *predictability*, as software process maturity increases the difference between targeted results and actual results decreases across projects. Therefore, organisations with more mature processes can expect to meet targeted dates with increased accuracy.
- With improved *control*, as maturity increases, the variability of actual results around targeted results decreases. For example, in organisations with immature software processes delivery dates for projects of similar size are unpredictable and vary widely. Similar projects in a higher maturity organisation, however, will be delivered within a smaller range.
- Improved *effectiveness* means that targeted results improve as the maturity of the organisation's software processes increases. In other words, as organisations mature their software processes costs decrease, development time becomes shorter, and productivity and quality increase.

The link between increased software process maturity and improved software development and maintenance performance has been confirmed by a number of studies, such as those performed by Herbsleb and Goldenson (1996) or the work of Lawlis, Flowe and Thordahl (1995).

In addition, a large number of case studies describing the tangible results of SPI initiatives as a whole have been published. (See Brodman & Johnson, 1996; Diaz & Sligo, 1997; Dion, 1993; Haley, 1996; Hersleb, Carleton, Rozum, Siegel & Zubrow, 1994; Jones, 1996; Pajerski & Waligora, 1997; SEI, 1997 for example.) Some of these results have been summarised by the Software Engineering Institute as follows (SEI, 1995):

"The typical return on investment, based on data from organisations that have done software process improvement for more than 3 years, is about 7:1, with an average gain in productivity of 37% per year, an average 18% increase each year in the proportion of defects found pre-test, an average 19% reduction in time-to-market, and an average 45% reduction in field error reports per year."

Finally, a number of **intangible benefits** have been ascribed to SPI initiatives. These include (SEI, 1995):

- Improved employee morale.
- Improved quality of work life.
- Fewer overtime hours.
- More stable work environment.
- Lower turnover of staff.
- Improved communication.
- Improved quality as reported by customers.
- Increased customer satisfaction.
- Increased competitive advantage.

1.3 Perceived Shortcomings of the SW-CMM and SPICE Models

The fact that SPI efforts based on models such as the SW-CMM and SPICE have, in many cases, resulted in significant improvements in software performance would seem to be evidence of the value and truth of these models. However, research in fields such as business process change or re-engineering and IS organisational transformation indicate that changes in an organisation's processes invariably require changes in other areas of the organisation in order to ensure that the objectives of the change are realised and that the change is sustainable.

For example, Kettinger and Grover (1995) describe the purpose of business process change, which incorporates radical process change (reengineering) and more incremental process improvement approaches (continuous improvement), as the transformation of business processes with the purpose of improving the process outcomes. They propose a descriptive model of business process change that is strategy-driven and based on environmental factors. The model contains subsystems, including business process, management, information and technology, people and organisational structures that produces outputs, including products and services. The full model is depicted in figure 1-1.

Kettinger and Grover (1995) go on to state that:

"Although it is possible that a BPR project may achieve breakthroughs in performance by only affecting one subsystem of the proposed process change model (e.g. a move from a hierarchical to a case-management structure), it appears more likely that the magnitude of the change will be amplified, and outcome impacts (both positive and negative) will be greater, as more than one subdimension is involved in the process change."

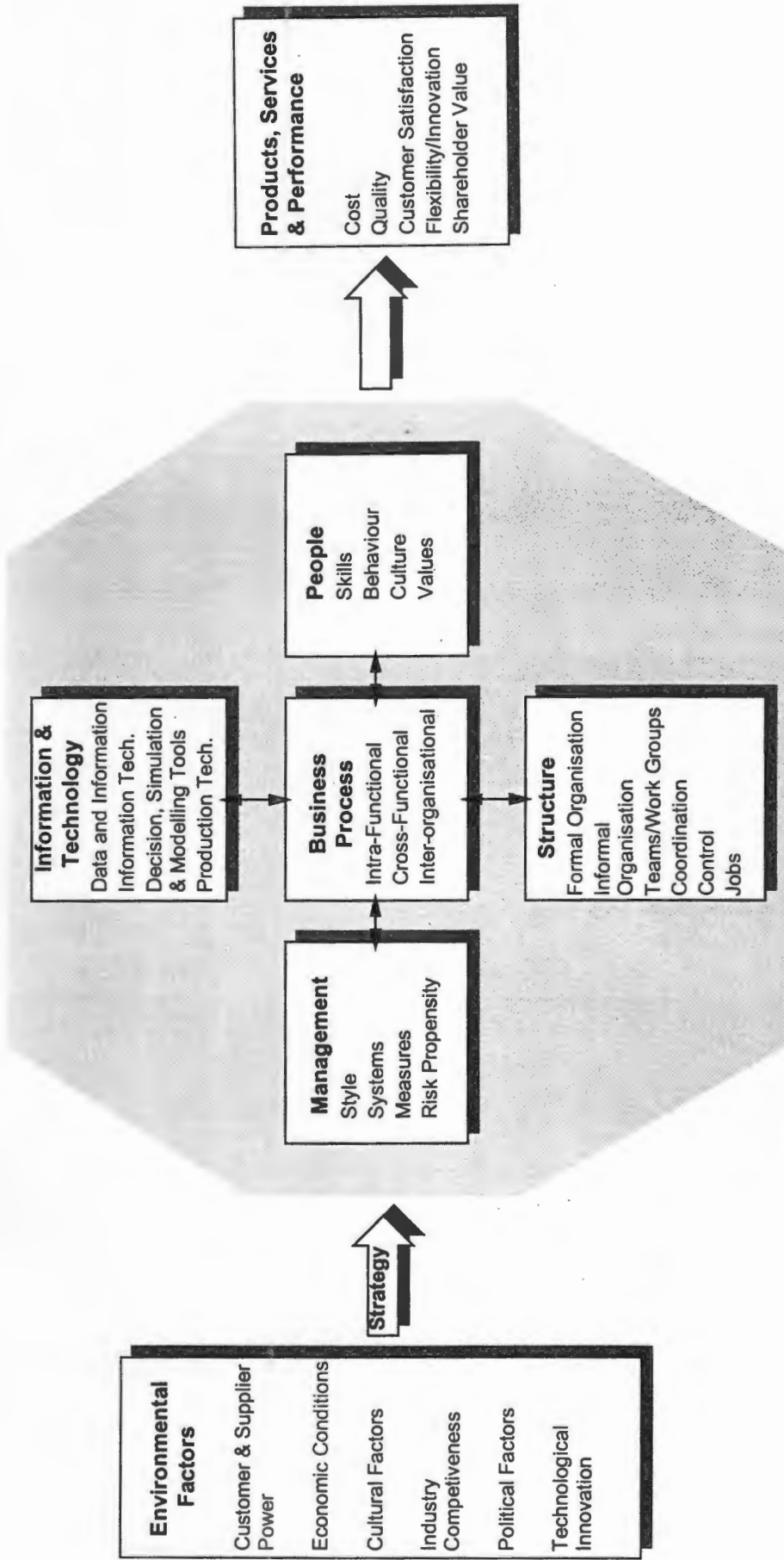


Figure 1-1: Business Process Change Model from Kettinger and Grover (1995)

IS organisational transformation is often undertaken by Information Systems organisations in response to the "acceleration of business change and the ubiquitousness of computing and telecommunications" (Boggis & Yarrington, 1997). It is an attempt to create the capability and capacity to satisfy the demands placed on them by their clients, in a cost effective way.

The notion that processes are but one aspect that should change when attempting to improve the performance of an IS organisation is evident in IS transformation research such as Clarke, Cavanaugh, Brown and Sambamurthy (1997); Rockart, Earl and Ross (1996); and Cross, Earl and Sampler (1997).

In a case study of an IS unit at Bell Atlantic, Clarke et al (1997) explore a new IS organisation design that helped transform the IS organisation into one with change-readiness capabilities. Change-readiness is defined as the "ability of an information systems organisation to deliver strategic IT applications within short development cycle times by utilising a highly skilled internal IS workforce" (Clarke et al, 1997). The new organisation design was described in terms of elements from a framework described in Galbraith (1995). This framework consists of a set of components, namely task, structure, processes, people skills and reward systems, that should be present and aligned with each other in order for an organisation to create a capability to realise its strategy.

Rockart, Earl and Ross (1996) conclude from their research into the future role of the IT organisation in a context of rapidly changing business and technology worlds that there are eight "imperatives" for those IT organisations that want to be successful in the late 1990's. These imperatives are listed in table 1-1.

Eight Imperatives for the IT Organisation
1. Achieve two-way strategic alignment.
2. Develop effective relationships with line management.
3. Deliver and implement new systems.
4. Build and manage infrastructure.
5. Reskill the IT organisation.
6. Manage vendor partnerships.
7. Build high performance.
8. Redesign and manage the federal IT organisation.

Table 1-1 Imperatives for the IT organisation from Rockart et al (1996)

Software process improvement falls into the realm of the seventh imperative, namely Build High Performance, and, to a lesser extent, into the area of imperative three, Deliver and Implement New Systems. The other imperatives address issues such as strategic alignment, partnerships with the business and other IT suppliers, leveraging new technologies and management systems, human resource development and changed organisational structures. Addressing a subset of the eight imperatives is not in itself sufficient to transform the IT organisation - "For an organisation to successfully use IT today, IT management must respond to the changing business and technology environment through effective efforts in each of the eight imperatives" (Rockart et al, 1996).

In a third example pertaining to IS organisational transformation, Cross, Earl and Sampler (1997) investigated the transformation of the IT function at British Petroleum over a period of six years. They conclude their case study by describing the British Petroleum IT transformation agenda in terms of Bartlett and Ghosal's (Bartlett & Ghosal, 1994; Bartlett & Ghosal, 1995; Ghosal & Bartlett, 1995) framework of Purpose, Process and People, stating that this provides a useful IS transformation agenda and an integrating framework. They go on to say that:

"We are not only proposing that any agenda for reform of the IT function should tackle all three constructs of purpose, process and people, but also suggesting that changes to just one set of dimensions of IT activities - purpose, process or people will be suboptimal without attention to the others. More precisely, the model implies that without careful consideration of purpose, a change agenda may be neither understood nor sufficiently ambitious. It proposes that process thinking and redesign is applied as much to the IT function as to the business at large. And it argues that inevitably transformation is achieved through people and thus people development is essential."

From the above work on business process change and IS organisational transformation it is arguable that in order to improve the performance of a process one generally needs to address a number of complementary areas.

A number of authors in the field of SPI share this conjecture in the context of software processes. For example, McGuire and Gibson (1997) state that "successful implementation of software process improvement efforts requires careful attention to a wide variety of technical, methodological, organisational, managerial, and process issues"; Hefley (1996) discusses the relationship between improving the software process and organisational practices; Downes, George and Chappell (1996) describe the relationship between improving the software process and technology; Curtis, Hefley and Miller (1995) mention that people, process and technology are all a part of an organisation's improvement efforts; and Perry, Staudenmayer and Votta (1994) have conducted research into organisational, social and technological factors could be considered when addressing the software process.

In a critique of the SW-CMM, Saiedian and Kuzara (1995) state that the SW-CMM "is not an exhaustive model or 'silver bullet'. It does not address several software management and engineering practices important to successful projects. For example, the CMM does not yet directly address expertise in a particular application domain, advocate specific tools, methods or software technologies; or address issues related to human resources (such as how to select, hire, motivate, and retain competent people). Neither does it address issues related to concurrent engineering, teamwork, change management, or systems engineering."

The authors of the Capability Maturity Model acknowledge the above, stating that although the SW-CMM "was deliberately created to focus on the software process", issues related to areas such as human resource management, people, technology and business planning must all be addressed (SEI, 1995).

Similar comments could be made about the SPICE model. Although it does explicitly address some engineering, management, systems engineering and human resource aspects that the SW-CMM does not, its focus and emphasis is still very much on the software process. It also does not cover all the improvement areas described above and could therefore be subjected to the same criticisms as the SW-CMM. (ISO, 1996a-i; Paulk, 1998)

In conclusion, it seems that despite the apparent successes realised from the application of SPI models, such as the SW-CMM and SPICE, overcoming the challenges of software development and maintenance may require more than a focus just on the software process. Other areas not emphasised by the SW-CMM and SPICE models probably also need to be considered in order to support an improvement effort and sustain its results.

It may therefore be valuable to extend the SW-CMM and SPICE models to take cognisance of the improvement context and complementary improvement areas. Doing so will leverage the strengths of these models, whilst at the same time help fill some of their gaps. This will be useful to practitioners wishing to improve the software performance of an IS organisation by addressing a wide range of improvement areas, not just those explicitly described by the SW-CMM and SPICE software process improvement models. It will also assist practitioners and researchers to place an SPI initiative in the context of other IS improvement actions.

The development and exploration of these proposed extensions to the SW-CMM and SPICE models is the focus of this research and the topic of the next section.

1.4 Proposed Extensions to the SW-CMM and SPICE Models

Figure 1-2 illustrates a proposed theoretical framework for extending the SW-CMM and SPICE models to explicitly consider changes other than those pertaining directly to the software process itself. The structure of this **SPI framework (SPIF)** is strongly influenced by Kettinger and Grover's (1995) work on the formulation of a theoretical framework of business process change (BPC) management and the SW-CMM and SPICE models. The usefulness of the Kettinger and Grover (1995) BPC framework for investigating SPI has been demonstrated in a study by Janz, Wetherbe, Davis and Noe (1997) on the impact of self-directed work teams on software development performance and employee satisfaction.

The SPIF is largely based on the notion that an SPI initiative is a special form of business process change, namely the incremental improvement of the IS organisation's "business processes", that is, its software processes (Janz et al, 1997; Davenport, Jarvenpaa & Beers, 1996; Deephouse, Mukhopadhyay, Goldenson & Kellner, 1996). The other significant factor that was considered is the layered architecture of the SW-CMM and SPICE models (SEI, 1995; ISO, 1996a-i). These SPI models define a series of process maturity levels that describe an evolutionary path, over time, from ad hoc, chaotic processes to mature, disciplined software processes¹. Each maturity level represents the attainment of certain process characteristics and is indicative of an organisation's software process capability. Different process attributes and improvement actions are emphasised at each maturity level.

The SPIF consists of three main components. The **Change Environment** describes the organisational environment and context for any proposed change or improvement initiative. The **Change Implementation** describes the change interventions and improvement actions carried out in order to achieve the objectives of the strategic initiatives. Thirdly, the **Change Outcomes** describe the results of the improvement actions.

¹ It should be noted at this point that the SW-CMM and SPICE models adopt slightly different philosophies regarding the definition and use of maturity levels. This is described in detail at a later point in the dissertation, but for the purposes of this discussion it suffices to say that the SW-CMM emphasises the overall process maturity of an organisation whereas SPICE evaluates the capability (maturity) of individual process instances (Paulk, Konrad & Garcia, 1995). Conceptually, however, the underlying process management concepts are sufficiently similar (Paulk, Garcia & Chrissis, 1996) for the differences to be ignored in the definition of the SPIF.

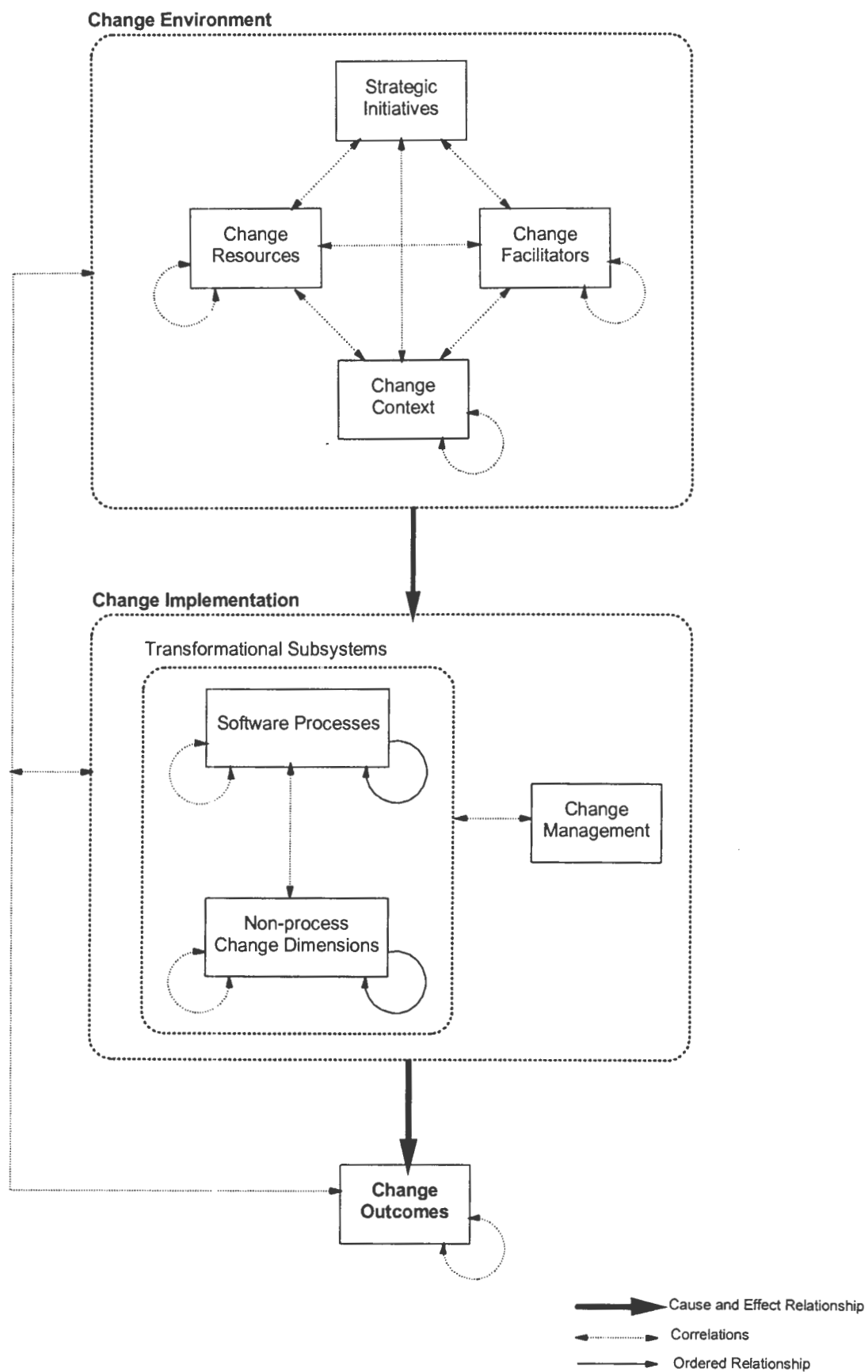


Figure 1-2: A Proposed Software Process Improvement Framework (SPIF)

The change environment component contains four elements. At the heart lie **Strategic Initiatives** where top management act as leaders in defining and communicating a vision for changing the organisation through SPI. This could entail envisioning, commitment and enabling from the senior management team, either in reaction to a need (e.g. poor performance), or as a proactive push to leverage potential opportunities (e.g. a higher software process maturity rating may enable an organisation to tender for contracts from which they otherwise might be excluded (Hunter, 1997, Rahardja, 1996)). In sum, strategic initiatives involve the formulation of a strategic plan, contribute to positive change outcomes through a specific plan of action, and motivate the entire software organisation towards achievement of the change goals. (Guha, Grover, Kettinger & Teng, 1997)

Strategic initiatives are constrained by the **resources** the organisation has at its disposal to realise the required change (e.g. funds for additional training), the existence of **change facilitators** in the organisation (e.g. cultural readiness and learning capacity (Kettinger & Grover, 1995)) and the **context** within which the organisation operates (e.g. the environmental context in terms of competitors or customer demands) (Robson, 1997).

The change implementation component proposes that a successful SPI initiative requires adjustments or improvements in the **non-process change dimensions** as well as in the **software processes**. Figure 1-1 shows some examples of these non-process change dimensions. The software processes and non-process change dimensions are collectively termed the "transformational subsystems" and represent those aspects of an organisation that can be changed in order to improve its performance. The changes in the transformational subsystems should support and mutually reinforce, or at the very least, not undermine, each other. The changes should also be supported by **change management** practices.

The software process element is modelled on the maturity levels of the SW-CMM and SPICE models. The SPIF therefore reflects a succession of software process maturity levels and makes allowance for the fact that at each maturity level the characteristics of the software process, the process management disciplines and the requirements for moving to the next level may be different.

The other change dimensions may also be subject to some form of maturity or staged growth model. The Software Engineering Institute has, for example, developed a People Capability Maturity Model (P-CMM) that provides a maturity framework, patterned after the structure of the SW-CMM, that focuses on continuously improving the management and development of the human assets of a software or information systems organisation (Curtis, Hefley & Miller, 1995). The SPIF, therefore, also suggests that improvements at each maturity level of the software process may require different interventions in the other dimensions.

Change management is critical to managing the resistance to change inherent in most people and to sustaining change within the organisation (Janzon, Nilsson, Gumpert & Sved, 1997). It is therefore useful to have a well-managed process of change to facilitate the implementation and institutionalisation of changes that are effected in of the transformational subsystems (Randall & McGuire, 1997).

The results of implementing a change initiative such as SPI are described in the **change outcomes** component of the SPIF. Kettinger and Grover (1995), for example, suggest that these outcomes can be measured at various levels. At the process level, the performance of the changed process (e.g. developer productivity, defect reduction, increased customer satisfaction) should be measured and compared to expected results. On the human level, however, many firms recognise that they can only achieve their objectives through people and, therefore, are placing employee quality of work life (QWL) issues high on their list of expected outcomes from change implementations. To quote Kettinger and Grover (1995):

"If human needs have been considered and change effectively managed, employees should experience improved working conditions in redesigned process tasks; this should increase employee job satisfaction and pride in work and strengthen their commitment to the organisation. Ultimately, this should make employees more productive in their jobs and better able to serve their customers".

The consideration of the SW-CMM and SPICE models in the SPIF implies another expected outcome, namely the establishment of a software environment that is able to apply continuous software process improvement and innovation in a rapidly changing business and technology world on a proactive and controlled basis with minimal disruption to the performance of the software process. This is the ultimate objective of the SW-CMM and SPICE models (SEI, 1995; ISO, 1996a; ISO, 1996b)

Since the framework recognises the incremental and continuous improvement nature of SW-CMM and SPICE based improvement efforts it is expected that change outcomes should be measured at appropriate intervals and that the change environment and change implementation areas take note of the lessons learnt.

In figure 1-2, the heavy, unidirectional arrows between the change environment and the change implementation, and between the change implementation and the change outcomes, indicate a cause and effect relationship. For example, the presence of the correct elements in the change environment could lead to successfully implementing change.

The bi-directional arrows represent possible correlations between different elements in the framework. For example, the software processes identified by the SW-CMM are represented by the "Software Processes" element in the SPIF. Process elements from other process improvement models, such as the Personal Software Process (PSP) (Humphrey, 1995) are also represented by this element. Different aspects of the different maturity models may be related in some way - the PSP may be used in conjunction with the SW-CMM (Humphrey, 1998a), or elements of the SW-CMM may be mapped onto the SPICE model's structure (Paulk, Konrad & Garcia, 1995). What the SPIF does not do at this stage is speculate on the nature of these correlations other than to say that they exist.

The unidirectional arrows on the transformational subsystem elements reflect an ordered relationship between different aspects of the same element. This is used, for example, to represent the evolution of the software processes through a successive series of maturity levels. This relationship could be seen as just another type of correlation between elements, but it is highlighted in the SPIF since the concept of a series of maturity levels is such a fundamental aspect of the SW-CMM and SPICE models.

In summary, the general thesis of the SPIF is that any significant SPI initiative requires a strategic initiative where top management act as leaders in defining and communicating a vision for change. (Although it is arguable SPI may be instigated by "people on the floor", the cost and duration of such an initiative is likely to make it sustainable unless it is ratified and supported by top management.) This strategic initiative and the appropriate change context, change resources and change facilitators must be supportive of each other and the SPI initiative. This, in turn, should facilitate the implementation of the required change which is effected through a balanced set of interventions in a number of change dimensions and a change management programme. The change implementation, along with the change environment, contributes to improved organisational performance, that is the change outcomes, which in turn could result in modifications in the change environment or change implementation.

1.5 The Purpose of the Research

The goal of this research is to propose and explore extensions to the SW-CMM and SPICE models that help overcome some of the perceived gaps in these models. The SPIF illustrated in figure 1-2 was used to guide this process and help find answers to questions such as:

- What should the change environment for SPI look like?
- What other change dimensions should be considered in conjunction with the software process to facilitate SPI?
- How should these change dimensions be aligned with the software process maturity levels described by the SW-CMM and SPICE models in the sense of being complementary and supportive of the goals of each maturity level.?

Taken from previous section - is it useful here: The SPIF illustrated in figure 1-2 is intended to be descriptive in the sense that it explains and categorises various aspects that could be considered in a software process improvement initiative.

There are five main objectives of this research. The first three objectives are concerned with the development of an Extended SPI Model (ESPIM) that is based on the structure of the SPIF. The fourth objective is to explore the validity of the ESPIM. The fifth objective is to identify future opportunities for research. Specifically the research has the following objectives:

1. To identify complementary improvement areas that the SW-CMM and SPICE SPI models do not necessarily address in detail as raised in the software process, business process change and IS organisation transformation literature. Loosely speaking, these improvement areas can be represented by any of the elements found in the change environment and change implementation components of the SPIF.
2. To determine how the identified improvement areas are related to the SW-CMM and SPICE models, and possibly to each other. It is possible that, at each software process maturity level, different complementary improvement areas, or different aspects of these areas, should be emphasised in order to achieve the purpose of that maturity level. For example, the IS organisational structure most suited to the earlier process improvement stages may be quite different from that required at a later stage.
3. To create a model, following the structure of the SPIF, from the information gathered above. The outcome of this is the Extended SPI Model (ESPIM) that reflects the SW-CMM and SPICE SPI models and the related complementary improvement areas. Here it fulfils the purpose of a descriptive model of a broad approach to SPI.

The ESPIM is intended to be a high level model and the detail included is limited to that

needed to understand the relationship between the SPI models and the aspects of those complementary areas that should be highlighted at each software process maturity level. It should also contain enough detail and evaluation criteria so that it can be used to guide case studies on the actual experiences of companies engaged in SPI initiatives. In this sense it is an evaluative model (Mashiko & Basili, 1997).

4. To conduct at least one case study in an organisation engaged in an SPI initiative in order to: a) determine the extent to which the ESPIM reflects the actual practice of an SPI initiative; b) identify reasons for deviations; and c) refine the model based on the findings of the case studies. The case study should also provide interesting insights into the motivation, progress, approaches, realities and lessons learnt from software process improvement initiatives in a South African organisation. The outcome of this is a very tentative exploration of the literature findings through empirical research.
5. To identify opportunities for future research. This research, being largely exploratory in nature, has the potential to open up a number of avenues for research. This may well turn out to be the greatest contribution of the dissertation.

The practical contributions of this research include:

- The enhancement of the SW-CMM and SPICE SPI models with those complementary improvement areas that are alluded to as being important to software process improvement, but which are largely ignored by the SW-CMM and SPICE.

- The development of an expanded understanding of how to improve the software development performance of an organisation by addressing a wide range of improvement areas, not just those explicitly described by the SW-CMM and SPICE models. This will be very valuable to practitioners wishing to improve their software development performance using a more holistic approach than that described by the current SW-CMM and SPICE models.
- The establishment of a basis for future research into the relationship between SPI and other improvement interventions that organisations may apply in an effort to improve their software development capability.

The rest of this dissertation describes the contributions from the literature to this topic, the development of the ESPIM based on the SPIF, and the findings of the case studies. It concludes with a summary of the main findings, limitations of the research and proposals for possible future research.

2. Literature Survey

The concepts proposed in the SPIF are explored in this chapter. Contributions from the SPI, business process change and IS organisational transformation literature are considered, firstly for the framework as a whole and then for each element in turn. The SW-CMM and SPICE models, particularly those aspects pertaining to the maturity levels, are explored in order to identify how other change dimensions can be aligned with the maturity levels. The chapter then concludes with the development of an ESPIM based on the structure of the SPIF and the contributions from the literature.

The scope of the ESPIM is limited to the elements of the SPIF and the potential mapping between the SW-CMM and SPICE maturity levels and the other change dimensions. The exploration of the other potential relationships between the elements is not addressed by this dissertation.

2.1 Overall Structure of the Proposed Framework

Over and above the research that was used in the development of the SPIF, a number of researchers have published findings that are supportive of its general structure.

Orlikowski (1993), for example, developed a theoretical framework for conceptualising the organisational issues around the adoption and use of CASE tools from grounded theory research into two organisations. The study characterised the organisations' experiences in terms of processes of incremental or radical organisational change. The results suggest that in order to account for the experiences and outcomes associated with CASE tools, researchers should consider the social context of systems development, the intentions and actions of key role players, and the implementation process followed by the organisation.

Figure 2-1 depicts Orlikowski's (1993) organisational change process around CASE tools. As can be seen, parallels with the main elements of the SPIF, namely the change environment, change implementation and change outcomes, are all clearly identifiable.

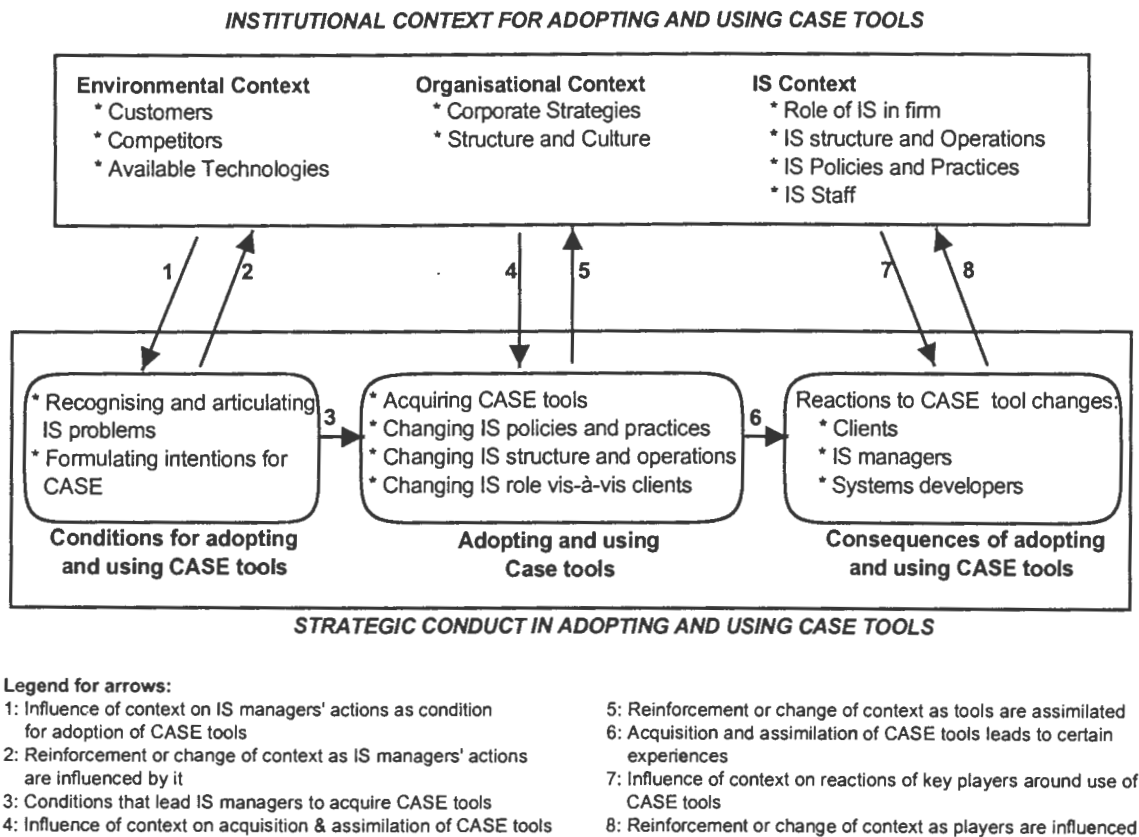


Figure 2-1: Organisational change process for the adoption of CASE tools (Orlikowski, 1995)

Although they do not propose a structured framework for SPI, Randall and McGuire's (1997) observations and conclusions, drawn from research on organisations that have SPI initiatives, contain many of the elements present in the SPIF. They note that "successfully implementing process improvement efforts requires attention to many organisational, people, process, quality, and methodological factors". Areas such as the processes, quality and improvement objectives, team structures, changing work modes and skills, management practices, change management and a change model, the organisational context, culture, learning capacity and strategy, outcome measurements; and the alignment between these factors should all be considered (Randall & McGuire, 1997). These areas are easily mapped to the framework's change environment, change implementation and change outcomes components.

The notion organisational areas other than the software process should be considered and aligned in order to facilitate and sustain change is well accepted in the process reengineering, and more recently, the IS implementation and transformation realms (Sarker, 1995; AlBanna & Osterhaus, 1998, Kettinger & Grover, 1995; Future Roles and Responsibilities for the IS Department, 1994; Rockart, Earl & Ross, 1996). Although different authors categorise the elements slightly differently, this "diamond of change" essentially consists of interrelated process, technology or information systems, management systems, organisational structure and skills, and culture components. The highlighted part of figure 1-1 represents Kettinger and Grover's (1995) version of this "diamond". The set of complementary change components is represented by the *software processes* and *non-process change dimensions* elements in the SPIF.

Therefore, an important consideration for an SPI initiative is that, at a macro level, all three components from the framework be explicitly considered and managed. Neglecting to give due consideration to the SPI initiative's environment, not addressing the implementation holistically, or failing to manage the outcomes, seems likely to lessen the impact of this sort of initiative.

2.2 Change Environment

The environment within which an SPI initiative is conducted has a significant impact on the achieved outcomes. The strategic direction of the organisation, the resources it has at its disposal to effect change, factors that facilitate the type of change required, and the context within which the organisation and its change initiative exist, are all factors that potentially determine the end result.

2.2.1 Strategic Initiatives

Zultner (1993) describes the importance of a correctly formulated strategy for improvement programmes to succeed. Such a strategy provides the means of aligning individual and team efforts with the organisation's strategic direction. The context of the Zultner (1993) paper is a TQM programme as applied to software development, but since the SW-CMM and SPICE models are really an application of TQM to the software process (SEI, 1995; Rout, 1995), the principles described here are applicable to the SPIF.

Zultner (1993) also explains that the organisational strategies must be vertically deployed throughout the organisation. In other words, the overall strategy of the organisation, or at the very least its intent, must be adopted by each department, team or unit and related to their particular work in a way that is supportive of and congruent to the overall strategy. In this way all areas and initiatives in the organisation are aligned and supportive of each other. Randall and McGuire (1997) reinforce this notion saying that "the goals, objective, values, beliefs, and actions of the organisation, management, teams and work units, and individuals should be kept in alignment throughout a process improvement effort".

This sentiment is further supported by Teng, Fiedler and Grover (1997) and Raffo (1997), who state that the extent to which an improvement initiative's strategy is congruent with the organisation's overall strategy and goals is a strong determinant of the initiative's success.

The Software Engineering Institute has proposed a model to guide the activities of an improvement programme based on the SW-CMM, the so-called IDEAL model (Gremba & Meyers, 1997). The IDEAL model begins with an initiation phase which emphasising the articulation of the business reasons for undertaking the improvement effort, irrespective of whether the stimulus is from unanticipated events or circumstances, an edict from senior management, information gained from benchmarking activities (Gremba & Meyers, 1997), or the efforts of a champion (Guha et al. 1997). Recognising the business reasons for the change can have a far-reaching influence on the effort's visibility, conduct, and ultimate success - change for the sake of change rarely results in significant improvement. Generally, when the business reasons for change are more evident, there is greater buy-in throughout the organisation and a greater chance for success (Gremba & Meyers, 1997).

Once the business reasons for SPI have been identified, the effort's contributions to business goals and objectives, its relationship with the organisation's other work, and the expected benefits should be identified. The support, commitment and sponsorship of senior management must be secured and made visible to the entire organisation. The sponsors must also, as part of initiating the SPI effort, enable it to proceed by committing the required resources (Gremba & Meyers, 1997). A number of authors comment on the importance of proper management sponsorship to the success of an SPI initiative (e.g. Wigle & Yamamura, 1997; Larner & Bray, 1997)

The formulation of a strategic initiative, either in reaction to a need or as a proactive push to leverage potential opportunities (Guha et al, 1997), defines the posture and sets the direction for an organisation; or, at lower levels, for an organisational unit or team. Adler et al (1992) suggest that such a strategy be expressed at three successive levels of detail. At the most general level a mission provides a clear sense of direction and purpose. The mission is then translated into measurable objectives that allow the function to continually assess its performance. Thirdly, the strategic plan identifies the path along which the function intends to meet its objectives and satisfy its mission.

Given the effort, cost and long duration of a typical SPI initiative (Gremba & Meyers, 1997; Brodman & Johnson, 1996; Jones, 1996), a clear strategic vision for the improvement effort is required for it to be sustained over time (Harkness, Kettinger and Segars, 1997).

Therefore, an SPI initiative should be motivated by a clear business need; be a defined strategic initiative which is aligned with the organisation's strategy; have clear, measurable objectives; have identified the actions required to realise the initiative; have top management take a lead in sponsoring and enabling the initiative; and have the everyone affected by the scope of the SPI initiative appreciate its purpose and be motivated to realise its outcomes.

2.2.2 Change Resources

SPI is seldom free. Although the benefits of a SPI initiative are potentially huge (see, for example, Dion, 1994; Jones, 1996), the changes required to improve an organisation's software processes can be very costly, not only financially, but also in terms of people's time and energy. Gremba and Meyers (1997), for example, state that SPI may involve 2-3% of the software organisation's people across a number of groups.

The ways in which an organisation's resources may be utilised to implement and sustain SPI are diverse. One of the most obvious examples is the establishment of a dedicated software engineering process group (SEPG) to facilitate the definition, maintenance, and improvement of the software processes used by the organisation (SEI, 1995).

Another example is the allocation of development staff to help define and implement the required software process changes. This has the effect of lessening the available resources for other projects. The impact on projects is also heightened by the fact that it is often the organisation's more skilled people that are allocated to the improvement initiative. Although this may be seen to have a negative effect in the short term, using the skills and knowledge of the organisation's best practitioners is seen to be critical to achieving widespread buy-in and the longer-term success of an SPI initiative. (Larner & Bray, 1997; Ahlgren, 1997)

Other examples of how resources may be applied for SPI include: the provision of widespread training; the utilisation of consultants to assist in the improvement efforts and perform assessments; assessment costs; the purchase of tools to support the improved processes or to assist in process management itself; the collection and management of process policies, procedures, standards and metrics; and the management and oversight of the improvement project itself (Larner & Bray, 1997; SEI, 1995; Dion, 1994; Haley, 1996). Given the typical long-term nature of SPI, the organisation must be prepared to commit these resources for a number of years, if not indefinitely.

Important considerations for SPI therefore include a reasonable understanding of the impact of SPI on the organisation's staff, an understanding of SPI costs up front - the development of a proper business case for SPI (e.g. McGibbon, 1996; Jones, 1996) is useful here - and a long-term commitment to improvement funds and resources from top management.

2.2.3 Change Facilitators

The availability of the required resources and a strategic commitment to SPI will not necessarily guarantee success unless the required changes are facilitated or enabled by a number of important factors. Kettinger and Grover (1995) identify four such facilitators. These are: the organisation's learning capacity, cultural readiness, knowledge-sharing capability (and leveraging IT to support this), and relationship balancing.

The first three facilitators seem to have direct application to SPI. Fichman and Kemerer (1997) describe a set of factors required to overcome "knowledge barriers" to the initiation and sustainability of software process innovations such as SPI. Choi and Behling (1997) highlight the ways in which the attitudes of top managers towards time, market and customers affect TQM programmes. Again, since SPI can be regarded as an instance of a TQM initiative, Choi and Behling's (1997) findings offer some useful insights into how management attitudes facilitate the changes required by SPI.

2.2.3.1 Learning Capacity

SPI entails continual improvement of the organisation's software processes. This is enabled by the organisation and its people acquiring and sharing knowledge, reflecting on lessons learnt, adapting to changes in their environment, and innovating. These are all aspects of a learning organisation (Guha et al, 1997; AlBana & Osterhaus, 1998), and an organisation that does not have this capacity to learn continually will struggle to succeed at SPI.

A number of approaches to achieving this learning have been suggested. Table 2-1 lists some of these from Guha et al (1997) and Randall and McGuire (1997), and proposes a few ways in which they can be realised in the context of SPI.

Learning Approach	Can be realised by:
Learning from other organisations	Utilising SPI models (e.g. SW-CMM, SPICE), industry standards (e.g. ISO12207), published case studies (e.g. Downes et al, 1996), participating in structures such as the SPI Network (SPIN) groups, utilising specialist consulting practices, etc.
"Learning by doing"	Piloting proposed process improvements on smaller projects, mentoring people in the use of new processes, etc.
Developing a knowledge base	Publishing standard processes and lessons learnt in a centrally available database.
Research and development	Utilising the Software Engineering Process Group as a software process R&D function, experimenting with new and innovative software tools and processes, etc.
Learning from past experiences	Utilising process metrics to provide insight into process performance and highlight opportunities for improvement. This is part of continual improvement and process innovation.
Training and Education	Sourcing training from in-house, specialist training organisations or tertiary institutions.
Team learning	Using collaboration and group work to solve problems and identify improvements (e.g. Harkness et al, 1997; Janz et al, 1997)

Table 2-1: Proposed learning approaches in the context of SPI

2.2.3.2 Cultural Readiness

Organisational culture can have a significant effect on the ability of an organisation to successfully learn and engage change and innovation (O'Reilly, 1989; Kettinger & Grover, 1995; AlBanna & OsterHaus, 1998). Organisational culture is often defined as the shared beliefs, values and norms in the organisation (Guha et al, 1997). AlBanna and Osterhaus (1998) say that culture is the dimension that binds people together within the organisation and includes the habits, values, constraints, inhibitions, aspirations, dreams and motivations that drive people to do what must be done.

SPI requires a cultural readiness for change and innovation. O'Reilly (1989) identifies a number of cultural norms that promote change and innovation, namely risk taking, openness, shared vision, respect and trust, high expectations for action, and a focus on quality. O'Reilly's (1989) research found that these norms were consistent across a wide range of industries and that they "all function to facilitate the process of introducing new ways of doing things and help people to implement them". It is therefore arguable that within the context of SPI, an organisation can be limited by the extent to which it is "culturally ready" for continual improvement and innovation, and that the presence of O'Reilly's cultural norms for change and innovation are important enablers of an SPI initiative.

2.2.3.3 IT Leveragability and Knowledge-Sharing Capability

Sensing, collecting, organising, communicating and using information are critical to the knowledge-based organisation and can have a direct impact on information flows, knowledge, culture, people and tasks. Guha et al (1997) note that IT often provides the means to accomplish the required management of information and that it can be harnessed to facilitate knowledge sharing. An organisation's ability to leverage IT in order to share knowledge can therefore have an impact on its ability to learn and change. Davenport and Beers (1995) also state that information about process characteristics, performance, and outputs is critical for process management since information suggests both the need for, and the direction of, potential process improvements, and aids in predicting and diagnosing problems with processes. Davenport and Beers (1997) go on to emphasise the positive role of IT in the collection and distribution of process information.

Examples of the types of information and knowledge that should be shared to enable SPI include defined processes and standards, process metrics, lessons learnt and improvement opportunities (Haley, 1996; Diaz & Sligo, 1997). IT that can be leveraged to help manage this information includes metrics collection tools and repositories (Haley, 1996; Diaz & Sligo, 1997), software process definition, instantiation and monitoring tools (Process Management White Paper, 1997), communications technologies such as groupware (Guha et al, 1997; Haley, 1996), and the Internet or Intranets to publish information (Wigle & Yamamura, 1997).

2.2.3.4 Overcoming Knowledge Barriers

Fichman and Kemerer (1997) describe some factors which can influence the propensity of an organisation to initiate and sustain the assimilations of software processes innovations. This research is based on the assimilation of object-oriented programming languages as an example of a software process innovation, which includes any technology or software practice that results in a change to the organisational processes used to develop software (Fichman & Kemerer, 1997). In this sense, it can be argued that SPI is also an instance of a software process innovation, albeit in a broader sense than Fichman and Kemerer use the concept.

Fichman and Kemerer (1997) showed through research in 608 IT organisations that software process innovations are more likely to be initiated and sustained in organisations where the burden of organisational learning is lower, either because they already possess much of the know-how and knowledge necessary to innovate, or because they can acquire such knowledge more easily or more economically. Specifically, it was shown that software process innovations are more likely to be initiated and sustained when organisations have a greater scale of activities over which the learning costs can be spread (learning-related scale), more extensive knowledge in areas related to the focal innovation (related knowledge), and a greater diversity of technical knowledge and activities in general (diversity).

The implication of Fichman and Kemerer's (1997) research for SPI is that organisations with the above characteristics are more likely be able to realise the changes and innovations required for SPI to succeed. This propensity to innovation follows from the fact that organisations with these characteristics can better amortise learning costs, can more easily acquire the knowledge needed to innovate, and have less to learn to begin with (Fichman & Kemerer, 1997). Fichman and Kemerer's research also showed that it is more often larger organisations that tend to have a higher learning-related scale, greater related knowledge and greater diversity of knowledge and activities. Therefore, larger organisations are more likely than smaller organisations to initiate and sustain an SPI initiative based on the SW-CMM or SPICE models. This is confirmed by the research of Downes et al (1996).

2.2.3.5 Positive Management Orientation

Choi and Behling's (1997) research indicates that top managers' underlying, often unspoken, orientations towards time, market and customers result in distinctly different approaches to TQM. These approaches, in turn, influence TQM's chances of success. Choi and Behling (1997) identify three different orientations of top managers: a developmental orientation, a tactical orientation and a defensive orientation. The basic characteristics of these orientations are summarised in table 2-2.

	Developmental	Tactical	Defensive
Main Concern	Growing the firm	Satisfying customer requirements	Surviving in a hostile world
Primary Time Focus	The future	The Present	The Past
Image of Customers	A partner	A demanding buyer	An opponent

Table 2-2: Top Managers' Orientations (Choi & Behling, 1997)

Choi and Behling (1997) concluded that there is a clear relationship between the orientation of top managers and the likelihood that their firms will have an active TQM programme. This relationship is the strongest where management has a developmental orientation. A developmental orientation is one in which management is a) mainly concerned with growing their business and client base by being a "world-class" software organisation, b) focused on improving current performance so that the organisation can be better positioned to compete in the future, and c) committed to seeing the customer as a partner in a co-operative relationship. The implication of Choi and Behling's (1997) research is that where top management has a developmental orientation the organisation has a far more comprehensive and consistent TQM programme than in other cases. In a similar vein, organisations with a developmental orientation tend to have far more successful TQM programmes.

Although the study had a limited scope, focused on a specific segment of the manufacturing industry, and was mostly concerned with TQM in the broadest sense of the term, the findings of Choi and Behling (1997) offer some insight into the impact that top management's orientation will on the success of an SPI initiative. Since the SW-CMM or SPICE models are really an application of TQM principles to the software realm (SEI, 1995), an SPI initiative based on these models is firmly rooted in the principles of TQM. It therefore follows that if the top management of a software organisation has a developmental orientation then, the organisation is more likely to be open to initiating and succeeding in an SPI initiative.

In summary, to facilitate the success of SPI it is important that the organisation is culturally and managerially ready to embrace change and innovation, that it has the will and ability to learn, and that this is supported by the means to effectively gather and share knowledge. Key indicators of this include cultural norms such as risk taking, openness, shared vision, respect and trust, high expectations for action, and a focus on quality. The organisation should also support the widespread acquisition, dissemination and sharing of software process knowledge; the means to identify, gather, analyse, and react on process outcomes; the learning-related scale, related knowledge and diversity required to foster innovation; and a management orientation that lends itself to software process improvement.

2.2.4 Change Context

The context or environment within which change is to occur has an effect on the formulation of strategic initiatives as the change context either provides the opportunities and stimulus for change or it acts as a barrier to effective change. Any strategic initiative of an organisation should be directed at exploiting the environmental opportunities and blocking the environmental threats in a way that is consistent with its internal capabilities (Robson, 1997). In the context of the SPIF, the internal capabilities of an organisation are represented by the change resources and change facilitators.

The work of Orlikowski (1993) on the adoption and use of CASE confirms that the institutional context within which the adoption occurred had an important impact on the change processes followed and the outcomes. Although Orlikowski's (1993) research focused on the adoption and use of CASE, it is arguable that her findings can be generalised to the SPIF as the implementation of both CASE and SPI are instances of innovation in the software organisation. Understanding the change context is therefore important in an SPI initiative.

Orlikowski (1993) provides a useful categorisation of the elements in the institutional context of a change or innovation in a software organisation. Table 2-3 depicts this categorisation of the institutional context into three main areas, each with a number of elements. The three categories are the Environmental Context, the Organisational Context and the IS Context.

Environmental Context	Customers Competitors Available Technologies
Organisational Context	Corporate Strategies Structure and Culture
IS Context	Role of IS in the firm IS structure and operations IS policies and practices IS staff

Table 2-3: Categorisation of the Institutional Context (Orlikowski, 1993)

Some instances of the impact of these elements from the change context on SPI are described below. For example, SPI is often a response by a software organisation to *competitive pressures* and *customer* demands. In some areas demonstrating a given software process maturity level is a pre-requisite for being able to tender for business as a provider of software. This is often a requirement in the United States of America and Europe, for example, when tendering for government contracts (Rahardja, 1996; Hunter, 1997). Proving a high software process maturity is also becoming more common in the commercial software arrangements, where companies are insisting on proof of software capability from their suppliers and partners (Walker & Knight, 1997).

The demands placed on modern businesses to be more competitive and responsive to client needs are also often a motivator for in-house software organisations to embark on SPI initiatives in an attempt to satisfy the business' software needs in a timely and cost efficient way (Larner & Bray, 1997; Clarke et al, 1997; Rockart, et al, 1996; Downes et al, 1996).

The proliferation of *available information technologies* and the rate at which they change puts pressure on the information systems departments of organisations that wish to leverage these technologies to support their *corporate strategies* or to gain competitive advantage (Ross, Beath & Goodhue, 1996). One of the objectives of a mature software organisation is the establishment of an organisational capability to continually and proactively adopt and utilise new technology in support of the organisation's objectives (SEI, 1995). Organisations that operate in an environment where it is necessary to continually exploit technology over the long term will therefore benefit from considering SPI.

There are cases where some *corporate strategies* may be naturally supported by an SPI initiative. For example, an organisation-wide TQM initiative would be well supported by an SPI initiative in the IS organisation (SEI, 1995). Corporate initiatives at Motorola, to radically improve the quality of products and reduce the time-to-market, have been complemented and supported by an SPI initiative in an area responsible for software development (Diaz & Sligo, 1997). The corporate strategies in the organisational context therefore have two impacts on a potential SPI initiative. In the first place, the wider corporate strategies will determine the appropriateness of an SPI initiative. Secondly, any SPI initiative should be aligned with the corporate strategies (Randall & McQuire, 1997; Zultner, 1993).

The impact of the organisational *culture* on SPI has been discussed previously. The *structure* and size of the organisation will have an impact on how easily information and knowledge can be shared and how easily the knowledge barriers associated with innovations can be overcome. These areas have also been discussed in detail previously.

IS often has an important *role* to play in an organisation as software is often a key enabler of the business, a source of competitive advantage or the purpose of business itself (Rockart et al, 1996, Ross et al, 1996). In these situations good software development practices are vital for the competitiveness of the organisation (Downes et al, 1996), and SPI must be an important consideration.

The *structure of IS* in an organisation can influence SPI in a number of ways. Again, a larger IS organisation may be more inclined to embrace SPI as it would be able to spread the costs of an SPI initiative and its required resources more widely. The way an IS organisation structures itself can also have a significant influence on the effectiveness with which it can carry out an SPI initiative. The use of centres of excellence, for example, have been shown to help improve learning, result in greater empowerment and innovation, and facilitate the development of specialised skills that can be economically leveraged across a number of projects (Clarke et al, 1997). This will greatly assist in achieving the goals of an SPI initiative. The availability of a strong support function in the IS structure can also enable an SPI initiative (Haley, 1996). Such a function can, for example, fulfil the role of a Software Engineering Support Group which facilitates the definition, dissemination and improvement of software processes and best practices.

Other factors which may predispose the organisation to SPI are strong *IS policies and practices* that emphasise quality and customer satisfaction (Zultner, 1993), a process view of the IS department's work (Deephhouse et al, 1996; AlBanna & Osterhaus, 1998), learning from past experiences, continuous improvement and innovation (AlBanna & Osterhaus, 1998), the use of policies, procedures and standards that encompass best practices (Haley, 1997), and management principles, guidelines, incentives and behaviour that encourage all of these practices (AlBanna & Osterhaus, 1998).

The *IS staff* background and training can often facilitate an SPI initiative. For example, Larner and Bray (1997) point out that an organisation which uses engineering processes in its main line of business and employs staff with an engineering education do not find the concepts behind SPI that hard to accept and apply. The organisation's ability to recruit good IS staff, provide ongoing skills development and training in support of its software processes and technologies, and utilise sound human resource management practices, has an important impact on its ability to improve its software processes in a sustained way (Curtis et al, 1995).

In summary, key indicators of an "SPI-friendly" change context include the presence of sound business imperatives for embarking on SPI, corporate strategies that support the exploitation of IT for business advantage, the presence of complementary corporate improvement initiatives such as TQM or productivity improvement drives, a culture that facilitates continual improvement and innovation, the resources and size to carry an SPI initiative, a dependence on software, IS organisational structures and resources that facilitate SPI, IS policies and practices that imply and support SPI, and an IS staff with the required skills and an inclination towards SPI.

2.3 Change Implementation

Realising improvements in an organisation's software development and maintenance performance probably requires changes in both its software processes and in other areas of the organisation that interact with these software processes. What is more, the changes made in each of these transformational subsystems must be aligned with each other in order to be most effective, otherwise, at worst, they can actually undermine each other. For example, attempting to establish a culture of team work can be hampered by a reward system that only recognises individual contributions. Transforming an organisation is a difficult endeavour at the best of times, and the application of a change management programme to facilitate the change and overcome change resistance is often critical to the lasting success of change initiatives.

2.3.1 Software Process Dimension

The crux of the SPIF is the improvement of software process performance. Models such as the SW-CMM and SPICE provide a means of assessing the *potential* of an organisation's software processes to perform in a certain way and also provide insights into what must be done to improve the processes so that their performance can be enhanced. This section explores these concepts more thoroughly in the light of the SPIF.

2.3.1.1 Software Process Improvement

Juran (1988) defines a process as a "systematic series of actions directed at the achievement of a goal". The Software Engineering Institute adheres to a similar definition in the context of software development and maintenance. It defines a software process as "a set of activities, methods, practices, and transformations that people employ to develop and maintain software and the associated products" (SEI, 1995).

One of the key assumptions of SPI models such as the SW-CMM and SPICE models is that the software processes of an organisation can be controlled, measured and improved (Humphrey, 1989). A controlled process produces software products according to plan, that is, it meets its goals. Measuring the software process helps understand the process's performance and gives insights into how it can be improved. Improving the software process increases the organisation's software process capability, or, stated differently, enables the attainment of higher goals, for example, decreased time-to-market for new products requiring software support, through the application of the software process.

Software process capability describes the range of expected results that can be achieved by following a software process, providing one with the means of predicting the most likely expected outcomes in future applications of the software processes on a software development or maintenance project (SEI, 1995). SPI, therefore, should increase the software process capability of an organisation in that it simultaneously improves the expected outcomes (results) and narrows the variance of performance (range).

An indicator of the capability of a software process is its maturity. A mature software process, in the Software Engineering Institute's view, is one that is defined, managed, measured, controlled, effective, consistently applied, and inclined to ongoing improvement (SEI, 1995). They contend that the more mature a software process is, the more capable it is likely to be. Conversely, they maintain that increasing the capability of a software process requires growth in the maturity of the software process. Software process maturity is therefore both necessary and sufficient for software process capability improvement.

Because the SW-CMM and SPICE models promote continuous process improvement based on many small, evolutionary steps rather than revolutionary innovations, it is useful to have a framework to indicate progress in increasing the maturity and capability of the organisation's software processes over time. Both the SW-CMM and SPICE models propose such a framework to organise these evolutionary improvement steps into a series of levels. These levels define an ordinal scale for measuring the maturity of an organisation's software processes and for evaluating its software capability. The SW-CMM calls the levels Maturity Levels, whilst the SPICE model refers to them as Capability Levels. (SEI, 1995; ISO, 1996b)

The SW-CMM model, defined by the Software Engineering Institute, and the SPICE standard both propose an approach that could be followed by an organisation to improve its software process capability. These SPI approaches are actually cyclical methodologies which can be applied to mature the organisation's software processes through successive maturity or capability levels. Figure 2-2 depicts the SPICE software process improvement steps (ISO, 1996g) and Figure 2-3 depicts the Software Engineering Institute's IDEAL approach (Gremba & Myers, 1997).

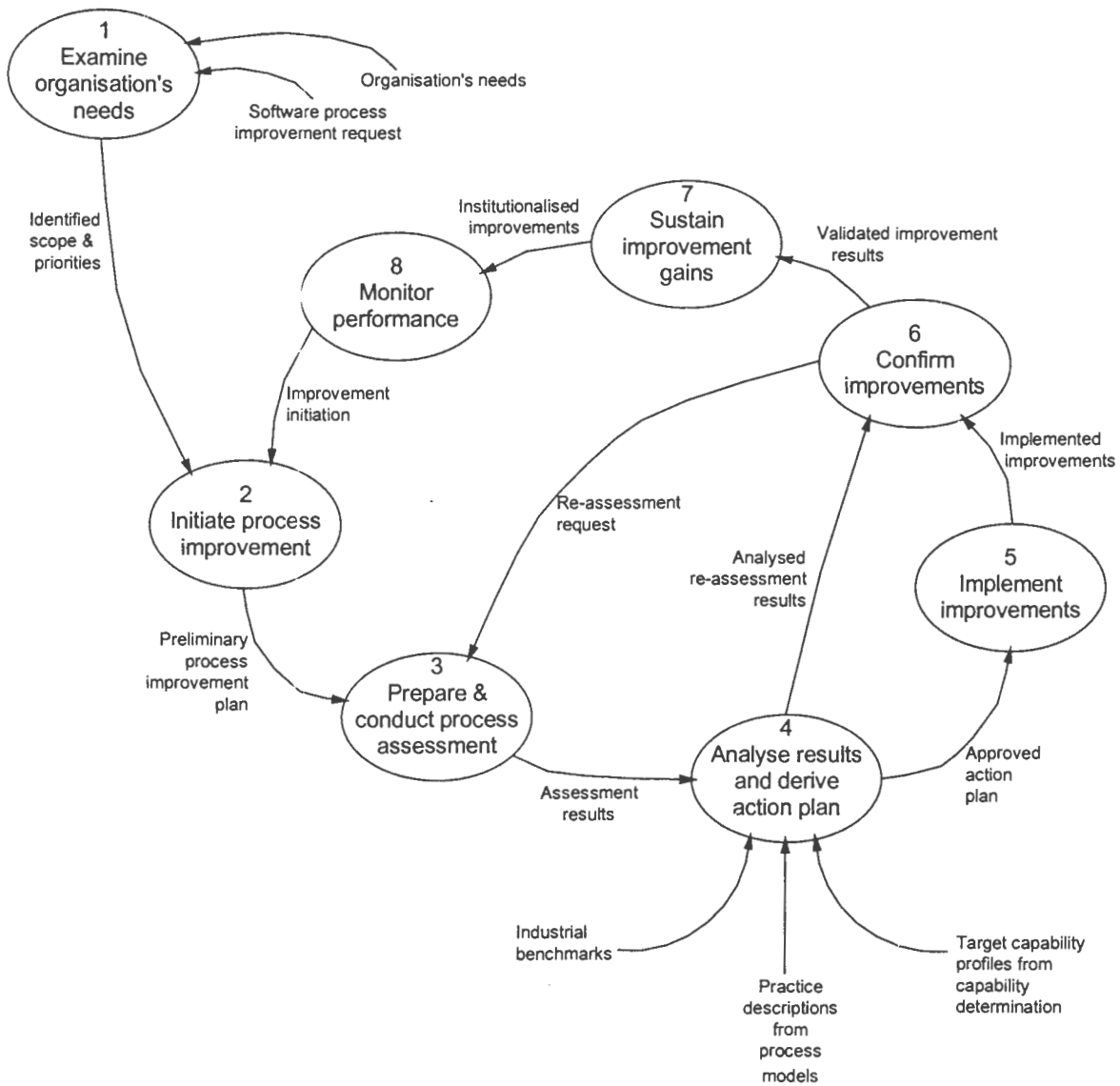


Figure 2-2: SPICE's SPI Steps (ISO, 1996g)

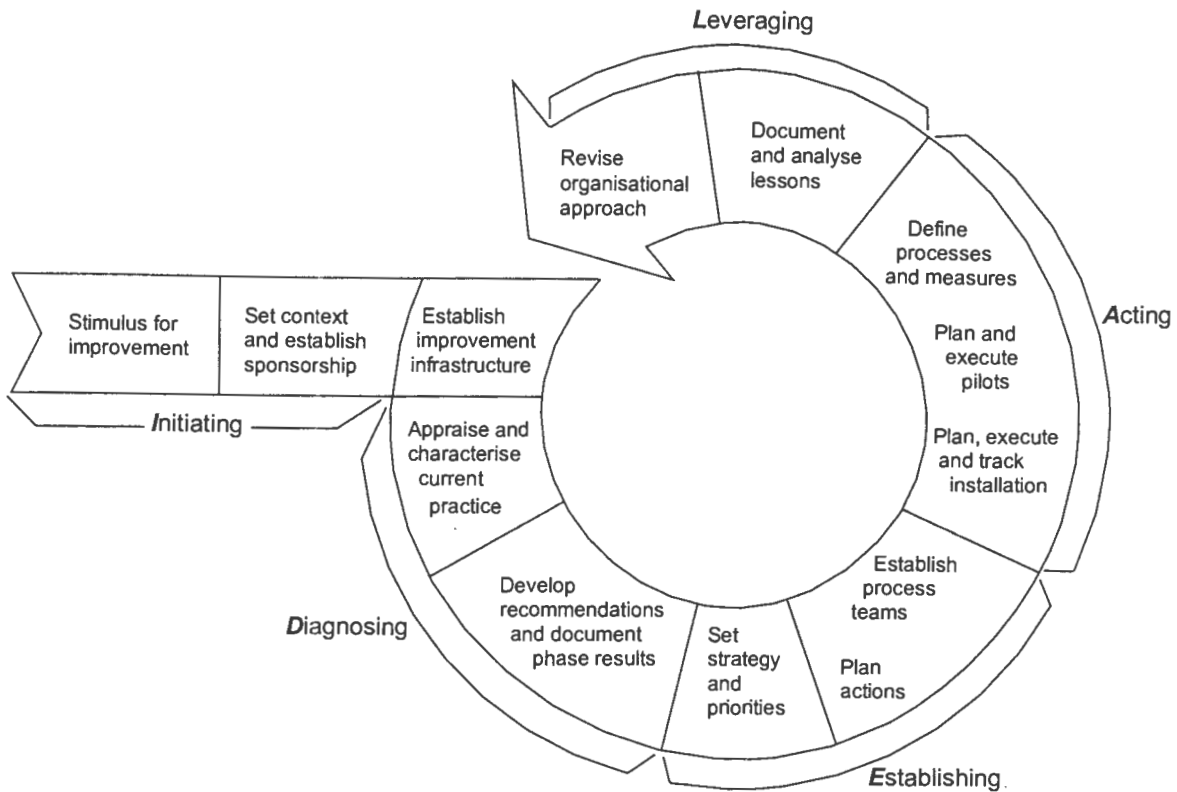


Figure 2-3: The IDEAL SPI approach (Gremba & Myers, 1997)

In conclusion, the repeated application of the above SPI methodologies over time, whether using IDEAL or SPICE, should improve the maturity of an organisation's software processes through the succession of maturity or capability levels. This increases the organisation's software process capability and results in improved software development and maintenance performance. The movement through a series of maturity or capability levels under the influence of SPI is represented in the SPIF (Figure 1-2) as the solid, uni-directional arrow on the software processes element. A discussion of the maturity or capability levels of the SW-CMM and SPICE models is included in the next section.

2.3.1.2 The SW-CMM and SPICE Models

The SW-CMM and SPICE models address four main areas of software process improvement (SEI, 1995; ISO, 1996b; Anonymous, 1997b). They identify and describe, at a high level, a number of *software processes*. In the case of the SW-CMM, only those processes which are deemed to contribute to directly increasing the software process capability of the organisation are described. SPICE, on the other hand, defines a complete set of software processes that are regarded as being essential to good software engineering. They define a maturity or capability level framework and a related evaluation system which are used to perform *software process assessments* to determine the *software process capability* of an organisation. The results of assessments and an understanding of an organisation's current software process capability are used to motivate and guide *software process improvement* activities. Figure 2-4 depicts these four elements and their inter-relationships.

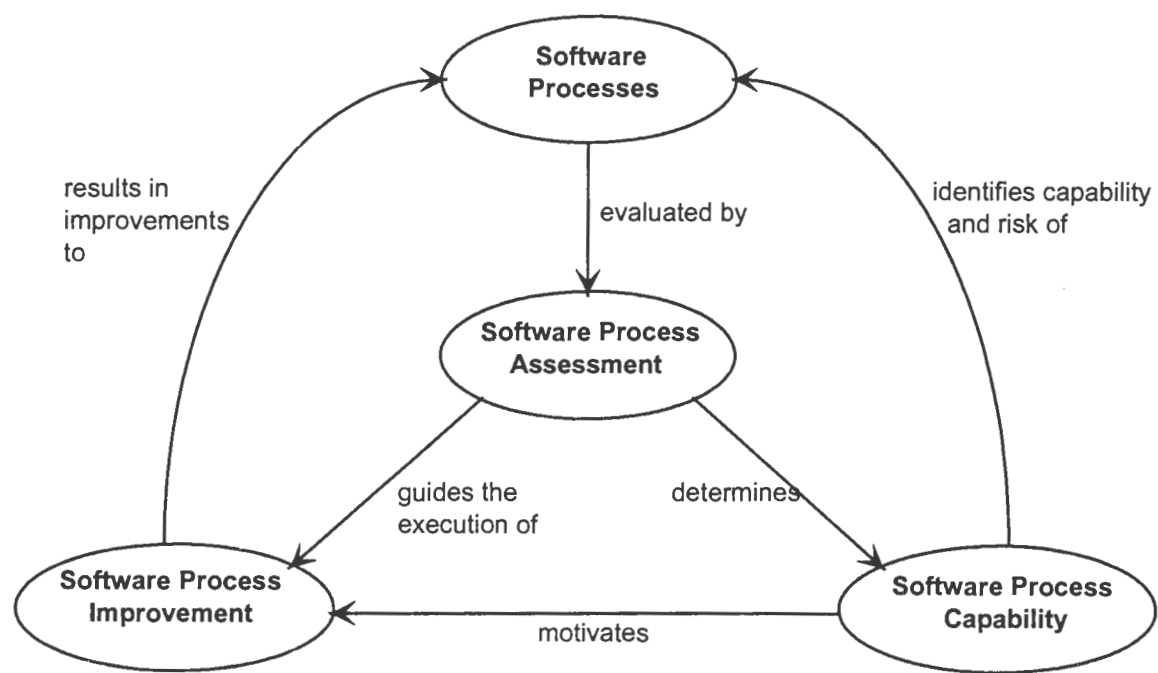


Figure 2-4: The main elements of the SW-CMM and SPICE models (adapted from Anonymous, 1997b)

The SW-CMM's maturity levels and SPICE's capability levels differ in the sense that the SW-CMM levels describe organisational process maturity, indicating the capability of the organisation's software processes as a whole. The SPICE model's capability levels, on the other hand, refer to the capability of individual processes. Despite this difference in approach between the models, their levels are conceptually similar (Paulk et al 1996), with each level representing the stabilisation of similar characteristics of the software processes. It is therefore possible to create a generic definition of the SW-CMM and SPICE models' various maturity or capability levels.

Table 2-4 provides such a generic definition. The names given to each level are loosely based on both the SW-CMM and SPICE models and should be seen to refer to the capability of a process instance in the context of SPICE and to the organisation's overall software process capability in the context of the SW-CMM. The changing nature of the software processes' outcomes as their capability increases over time is highlighted in the right-hand column of the table.

The SW-CMM and SPICE levels are cumulative in that each successive level builds on the previous levels. In other words, a process rated at a higher maturity or capability level must satisfy all the requirements of that level and all the preceding levels.

Increasing Software Process Maturity & Capability			Risk Waste	Quality Productivity
Level	Focus	Definition		
0 Incomplete	None	The process is not implemented or fails to achieve its purpose. This level is not defined in the SW-CMM.		
1 Ad hoc or Performed informally	Competent people and heroics	The processes are performed in an ad hoc manner, often without rigorous planning or tracking. Quality and schedule targets are frequently missed and success is based on individual knowledge and effort rather than on organisational infrastructure.		
2 Repeatable or Managed	Project management processes; Basic process control	Process performance is planned and tracked in terms of cost, schedule and quality. Standards and requirements for deliverables of the processes are defined and verified, and corrective action is taken when necessary. At this level the basic process discipline is in place to repeat earlier successes on projects with similar applications.		
3 Defined or Established	Organisational process definition and implementation	Processes that support the business goals of the organisation and are based on good software engineering practice are documented, standardised and implemented (required infrastructure and training is in place) throughout the organisation. All projects use approved, tailored versions of the organisation's standard processes.		
4 Quantitatively Managed or Predictable	Product and process quality; Process measurement and control	Process goals are established and detailed measures of the software process performance and work product quality are collected and analysed. This leads to a quantitative understanding of process capability and an improved ability to predict performance. The process and its deliverables are objectively managed, understood and controlled.		
5 Optimising	Continual process improvement & innovation	Quantitative process effectiveness and efficiency goals (targets) for performance are established based on the business goals of the organisation. Continuous process improvement against these goals is enabled by quantitative feedback from performing the defined processes and from piloting innovative ideas and technologies.		

Table 2-4: Generic Maturity Level Definitions (adapted from SEI, 1995 and ISO, 1996b)

In the SW-CMM model the maturity levels are described in terms of *key* process areas (KPA's) which indicate where an organisation should focus to improve its software processes and identify the issues that must be addressed to achieve a maturity level. The adjective "key" indicates that there are processes that are not considered fundamental to achieving a maturity level and only the process areas that have been identified as key determinants of process capability are described in the SW-CMM. Processes that are used to develop and maintain software, but do not necessarily contribute directly to maturing the organisation's software processes are therefore not described by the SW-CMM (SEI, 1995). Table 2-5 summarises the SW-CMM, depicting the maturity levels and assigned KPA's. Note that each KPA is assigned to a single maturity level in the SW-CMM, although it will continue to be applied and assessed at subsequent levels.

Maturity Level	Key Process Areas
1 Initial	None
2 Repeatable	Requirements Management Software Project Planning Software Project Tracking & Oversight Software Subcontract Management Software Quality Assurance Software Configuration Management
3 Defined	Organisation Process Focus Organisation Process Definition Training Programme Integrated Software Management Software Product Engineering Intergroup Coordination Peer Reviews
4 Managed	Quantitative Process Management Software Quality Management
5 Optimising	Defect Prevention Technology Change Management Process Change Management

Table 2-5: Key process areas of the SW-CMM by maturity level

The SPICE model differs from the SW-CMM in that it describes a universal set of software engineering and management processes (not just a set of key process areas). It also describes a set of process management attributes and activities which are grouped together to form capability levels. As can be seen in table 2-6, the SPICE processes are grouped into five main process categories.

Process Category	Processes
Customer Supplier	Acquire Software Manage Customer Needs Supply Software Operate Software Provide Customer Service
Engineering	Develop System Requirements and Design Develop Software Requirements Develop Software Design Implement Software Design Integrate and Test Software Maintain System and Software
Support	Develop Documentation Perform Configuration Management Perform Quality Assurance Perform Work Product Validation Perform Work Product Verification Perform Joint Reviews Perform Audits Perform Problem Resolution
Management	Manage the Project Manage Quality Manage Risks Manage Subcontractors
Organisation	Engineer the Business Define the Process Improve the Process Provide Skilled Human Resources Provide Software Engineering Infrastructure

Table 2-6: SPICE process categories and processes

The capability of each individual software engineering or management process in the SPICE model can be determined by its attributes and the extent to which it adheres to the specified process management activities. The process management attributes and activities address some aspect of process implementation, institutionalisation or effectivity. Table 2-7 shows how the process management attributes and activities are grouped into capability levels. Note that SPICE capability levels have different names to the SW-CMM maturity levels. The two models should be compared by looking at the number of the level.

SPICE processes are not assigned to a specific capability level, but grow in capability as the relevant process management activities are applied to them. This aspect of the SPICE model's architecture makes it possible to assess the maturity of an individual process or process area, whereas the SW-CMM emphasises the assessment of the capability of the organisation's software processes as a whole. It is, however, possible to determine the organisational software process maturity from a SPICE assessment. In this case the organisational maturity is defined as a set of profiles for all the processes.

Capability Level	Process Attributes and Management Practices
0 Incomplete	None
1 Performed	Process Performance Attribute <ul style="list-style-type: none"> • Ensure base practices are performed
2 Managed	Performance Management Attribute <ul style="list-style-type: none"> • Identify Resource Requirements • Plan the performance of the process • Implement the defined activities • Manage the execution of the activities Work Product Management Attribute <ul style="list-style-type: none"> • Identify integrity and quality requirements • Identify integrity and quality activities • Manage configuration of work products • Manage the quality of work products
3 Established	Process Definition Attribute <ul style="list-style-type: none"> • Identify the standard process definition • Tailor the standard process • Implement the defined process • Provide feedback Process Resource Attribute <ul style="list-style-type: none"> • Define human resource competencies • Define process infrastructure requirements • Provide adequate skilled human resources • Provide adequate process infrastructure
4 Predictable	Process Measurement Attribute <ul style="list-style-type: none"> • Define process goals and associated measures • Provide resources & infrastructure for data collection • Collect the specified measurement data • Evaluate achievement of process goals Process Control Attribute <ul style="list-style-type: none"> • Identify analysis and control techniques • Provide resources & infrastructure for analysis & process control • Analyse available measures • Identify deviations and take required control actions
5 Optimising	Process Change Attribute <ul style="list-style-type: none"> • Identify and approve changes to the standard process definition • Provide adequate resources to implement changes • Implement the approved changes • Validate the effectiveness of process change Continuous Improvement Attribute <ul style="list-style-type: none"> • Identify improvement opportunities in a systematic & proactive way • Establish an implementation strategy • Implement changes • Validate the effectiveness of process change

Table 2-7: SPICE capability levels, process attributes and management practices

The SW-CMM is said to have a "staged" architecture, whereas the SPICE model has a "continuous" architecture (Paulk et al, 1996). Given its staged architecture, the SW-CMM model can be described as (Paulk et al, 1996):

- an *organisation-focused* model, since its target is the organisation's process capability,
- a *descriptive* model, because it describes organisations at different levels of achieved capability,
- a *path* model, because it describes a software process improvement path through the terrain, and
- a *prescriptive* or normative model, since it prescribes in general terms how an organisation should improve its processes.

The SPICE model, having a continuous architecture, can be described as (Paulk et al, 1996):

- a *process-focused* model, since its target is process capability,
- a *terrain* model, from the analogy to a description of the software process terrain, and
- a *reference* model, since its primary use is in assessment as the reference for rating purposes.

Paulk et al (1995, 1996) further differentiate between the SW-CMM and SPICE models by identifying the advantages and disadvantages of their staged and continuous architectures. Table 2-8 summarises their assessment, which is a useful aid in understanding the differences between the SW-CMM and SPICE models.

	SW-CMM's Staged Architecture	SPICE's Continuous Architecture
Advantages	<ul style="list-style-type: none"> • It explicitly describes organisational capability in terms of maturity levels. • It focuses organisations on the "vital few" issues in process improvement. • It identifies improvement priorities that are generally true for any (software) organisation; that is, it provides an organisation with an overall improvement path. E.g. the SW-CMM prescribes addressing project management issues before engineering ones. • The guidance in the key practices and sub-practices of the key process areas is a significant help in understanding what a key practice or goal means. • A SW-CMM assessment potentially requires less effort than a SPICE assessment. 	<ul style="list-style-type: none"> • It fully describes the evolution of processes from ad hoc to continuously improving. • It measures the evolution of each process separately, allowing an improvement team to view its aspects independently of other processes and provides greater granularity of measurement and analysis. • Adding processes and integrating with other models is simply a matter of defining the purposes and basic practices of the processes and applying the process management attributes and activities for rating the processes. • The maturity levels can be layered on top of the continuous model's capability levels.
Disadvantages	<ul style="list-style-type: none"> • It only provides a snapshot description of the evolving process, people can lose sight of the processes that are not the focus of a particular maturity level. • It may be difficult for the non-expert to tailor or extend the SW-CMM. 	<ul style="list-style-type: none"> • It does not explicitly describe organisational capability. • It does not prescribe any particular organisational improvement path. • Less critical process issues can drown out the "vital few" when there are clashes over improvement priorities. • The number of rating decisions is large: over 1000 for a full-blown SPICE assessment. • Given the higher level of abstraction of the generic SPICE processes, it can be difficult to interpret what they mean or what is required by them.

Table 2-8: The advantages and disadvantages of the SW-CMM and SPICE architectures (adapted from Paulk et al, 1995 & 1996)

Finally, although the organisational software capability, as described by the SW-CMM, and process capability, as described by SPICE, are "inextricably intertwined", building a continuous model on top of a staged architecture is very difficult because of the fact that the staged model only allocates a process to a single level. On the other hand, building a staged model on top of a continuous architecture is much easier to do. Paulk et al (1995) demonstrate how the maturity levels of the SW-CMM can be layered on top of the SPICE capability levels at a high level of abstraction (figure 2-5). The fact that the SW-CMM version 1.1 does not cover the customer-supplier relationship in detail is depicted by only a partial shading in the Customer Supplier category.

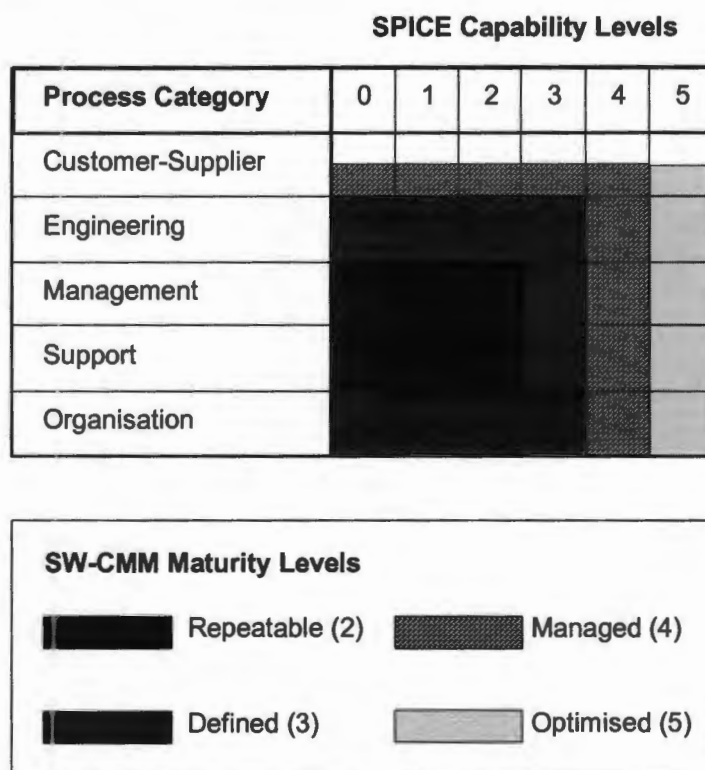


Figure 2-5: Mapping the SW-CMM maturity levels to SPICE capability levels

The mapping depicted in figure 2-5 is useful for organisations wishing to compare or use the two models them in conjunction with each other. For example, organisations may wish to use the SPICE model because of its more comprehensive set of software processes, its flexibility and its extensibility. However, they may also utilise the SW-CMM for its guidance in prioritising improvement areas. Alternatively, an organisation may have selected to use one of the models, but be required by their customers to be assessed using the other model. Here again, understanding the mapping between the SW-CMM and SPICE models will be an useful aid. Finally, this mapping is accommodated in the SPIF of figure 1-2 by the bi-directional arrow on the software processes element.

2.3.2 Non-Process Change Dimensions

As described in the SPIF, transforming an organisation's processes requires deep changes in the "key behaviour levers" of the organisation. These levers can include jobs, skills, information technology, organisational structures, measurement systems, culture, etc. The organisation's circumstances, motivations and change objectives should determine which "levers" are appropriate at any point during a change initiative. (Stoddard & Jarvenpaa, 1995).

This section identifies a number of change dimensions that can complement the SPI and it attempts to align these with the generic description of the SW-CMM and SPICE models' maturity or capability levels as proposed in section 2.3.1.2. The change dimensions are categorised into the four main groups identified by Kettinger and Grover (1995) and depicted in figure 1-1, namely *structure*, *people*, *information and technology*, and *management*. Other areas that may impact an SPI initiative, but do not fit into the above categories, are described in the 'Other Enablers' section.

The alignment between the identified non-process change dimensions and the maturity or capability level that they support is performed from the perspective of the software processes. For example, a particular dimension may be assessed as strongly supporting the attainment of Level 3 and is therefore aligned with that level. As is the case for the software process maturity levels, and unless stated otherwise, any dimension is regarded as being relevant to the level at which it is mapped as well as to all subsequent levels. No attempt is made to take a reverse look and answer questions such as how a given process maturity may support the dimensions or whether the dimensions can add value in their own right.

Also, unless stated otherwise, a mapping to a level is regarded to apply to both the SW-CMM and SPICE models. This follows from the fact that the purpose and intent of the levels in the two models, although described in different ways, are very similar. The generic maturity level definitions of table 2-4, the SW-CMM key process areas of table 2-5 and the SPICE process management practices and attributes of table 2-7 were examined and contrasted to the each dimension in order to determine how to align a specific non-process change dimensions to the maturity or capability level that they support.

2.3.2.1 Structure

With regards to structure, three main areas are addressed. These are the use of changed organisational designs such as centres of excellence, a focus on team-based practices, and organisational structures to enable an SPI initiative itself.

a. Changed organisational designs

Truly process-based organisations may find that they need to change their structures to aid the management and development of people. *Centres of excellence* provide such an organisational structure (CSC, 1994). A centre of excellence is an organised group of people who share a specific disciplinary affiliation and competence. In implementations of this organisational structure, staff are assigned to centres of excellence which are focused on skills or people development, usually under the leadership of a coach or mentor. The actual performance of work is not done in the centres of excellence. Rather, people with the correct skills mix are assigned from the centres of excellence to work on a "process instance" which delivers the work. "Process instances" are usually managed as projects under the leadership of a project manager. (Clark et al, 1997)

The benefits of centres of excellence include the separation of people development from the context of short-term, delivery focused projects; a focus on the development and renewal of skills; the ability to define jobs around competencies; the ability to allocate scarce resources to projects across the organisation in a way that is more flexible and responsive to change; and the opportunities it gives employees to work in a wider variety of areas in the organisation (Clarke et al, 1997; CSC, 1994).

The use of centres of excellence makes the most sense in a context where the organisation has project management practices in place, where processes are defined and widely used across the organisation, and where roles and training needs are based on the defined processes. Project management is usually stabilised at level 2, but the other process characteristics that would benefit from centres of excellence are only realised at level 3. Therefore, although centres of excellence can be useful at level 2, they are most strongly aligned with level 3.

b. Team-based practices

Software development teams and team-based development processes are frequently instituted in response to the complexity and size of today's systems (Hefley, 1996). They are also often cited as a characteristic of successful process and quality environments (Randall & McGuire, 1997). Although the SW-CMM addresses issues such as inter-group co-ordination and training, it does not explicitly address team-related issues (Hefley, 1996). The SPICE model also just touches on team issues in the "provide skilled human resources" process with base practices for training, definition and empowerment of project teams (ISO, 1996e).

The People Capability Maturity Model (P-CMM) (Curtis, Hefley & Miller, 1995), briefly introduced in section 1.4, has two key process areas (KPAs) that explicitly address team issues: the *Team Building* KPA and the *Team-Based Practices* KPA. The team building KPA seeks to capitalise on opportunities to create teams that maximise the integration of diverse knowledge and skills. It also addresses aspects such as matching potential team members to the knowledge and skill requirements of the team, training all new team members in team skills, defining objectives for team performance, tailoring standard processes for use by the team, and periodically reviewing performance of the team.

The team-based practices KPA of the P-CMM is designed to tailor the organisation's workforce practices to support the development, motivation and functioning of teams. This includes ensuring that the work environment supports team functions, setting performance criteria and reviewing team performance, involving team members in performing workforce practices such as recruitment and performance management, and reflecting team criteria in individual compensation decisions.

Janz et al (1997) explored the impact of self-managed teams or *self-directed work teams* (SDWTs) on improving the performance of IS organisations. SDWTs are groups of co-located workers who are typically multi-skilled, self-regulate work on interdependent tasks, and have a large degree of autonomy in how they function. The study concluded that the presence of SDWTs results in improvements in both the work processes of the IS organisation and the quality of work life of the employees.

A software organisation could therefore create an environment, and set of processes and practices, that facilitate the establishment of self-directed work teams for projects that instantiate the organisation's software processes. The teams that are created in this context may be staffed from the centres of excellence if they exist.

SDWTs often have control over their own work methods, schedules and the assignment of members to tasks, and are accountable for their own performance, including the quality of the work they deliver (Janz et al, 1997). They will support the development and maintenance of software at all maturity levels and are therefore regarded as being appropriate at all levels. The practices of SDWTs can be greatly enriched by the Team Building and Team-Based Practices KPAs of the P-CMM in organisations where, from level 3 onward, well-defined processes are used in all areas, are staffed by appropriately skilled and trained personnel, and have proper co-ordination between groups. In the SW-CMM the relevant level 3 KPAs are Organisation Process Definition, Training Programme, Integrated Software Management and Intergroup Co-ordination. The SPICE Process Definition and Process Resource attributes are both relevant, especially since they are supported by the Define the Process and Provide Skilled Human Resources processes.

c. Organisational structures to enable SPI

A number of formal and informal structures may be put in place to enable an SPI initiative and to enable the practice of the processes prescribed by the SW-CMM and SPICE models. Figure 2-6, adapted from Orville (1998) and ISO (1996g), depicts some of the process improvement team relationships.

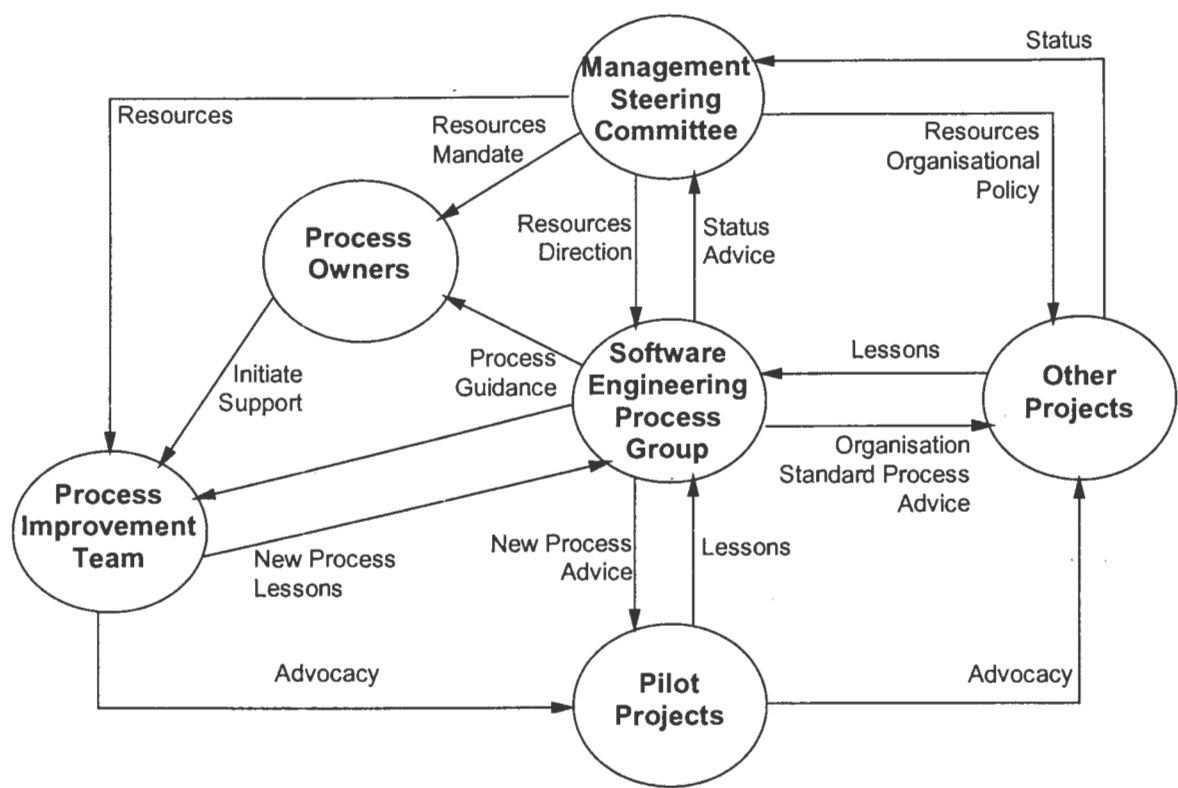


Figure 2-6: Process Improvement Team Relationships (adapted from Orville, 1998 and ISO, 1996g)

The *Management Steering Committee* sponsors an SPI initiative and must show visible commitment to SPI, create an environment conducive to SPI, provide adequate resources and funding, insist on adherence to the software process policies and standards, monitor the progress of the initiative, and be accountable for the long-term success of the SPI initiative (ISO, 1996g; Lerner & Bray, 1997; NRaD Software Process Improvement Charter, 1996).

A *Process Owner* is responsible for the processes overall performance and must co-ordinate all process functions and improvements to ensure cost-effective customer and internal personnel satisfaction (Harkness et al, 1996; ISO, 1996g; Larner & Bray, 1997). The Process Owner therefore initiates and supports process improvement actions and monitors the outcome. Given the scope and functions of a process owner, it is arguable that this person should be a relatively senior person in the organisation with some responsibility for the performance of the processes that they own.

Process improvements are frequently effected by *Process Improvement Teams* (Harkness et al, 1996; Orville, 1998; ISO, 1996g). These teams typically consist of experienced practitioners and process improvement experts, and are formed to carry out discrete process definition or improvement tasks. They are generally temporary structures that are disbanded once their task is completed. The process definitions and improvements effected by the Process Improvement Teams are usually tested and refined on pilot projects before being deployed on projects throughout the organisation.

The final grouping is the *Software Engineering Process Group* (SEPG). This is a group of process experts who facilitate the definition, maintenance and improvement of the organisation's software processes and provide specialist support, guidance and advice to the Management Steering Committee, the Process Owners, the Process Improvement Teams and projects applying the processes (SEI, 1995; Humphrey, 1989). Other responsibilities of the SEPG include performing assessments, sourcing training for the organisation's defined software processes and acting as the focal point or repository for the organisation's defined software processes (NRaD Software Process Improvement Charter, 1996; SEI, 1995).

The Management Steering Committee, Process Owners and Process Improvement teams are appropriate at all maturity or capability levels. Without management commitment and people to own and improve the different processes, an SPI programme will never get off the ground. As the organisation's processes mature, however, the focus of these groups will change. For example, at level 1 they will emphasise the basic concepts that are needed to perform the work and will tolerate differences between projects. As they move towards level 3 they will emphasise the definition of a common set of processes across the organisation. At level 4 they will focus on using measurements to remove causes of deviation in the process usage, and at level 5 they will focus on defect prevention and proactive improvements. The three groups are therefore assigned to level 1 onwards.

The SEPG can have a role to play at all levels, especially if it is given the responsibility for co-ordinating the overall SPI initiative. However, the presence of the Management Steering Committee and Process Owners makes this group less crucial at levels 1 and 2. It is, however, required from level 3 onwards where it co-ordinates the use and improvement of the common defined processes across all areas. A SEPG is also a requirement of the Organisation Process Focus KPA of level 3 of the SW-CMM (SEI, 1995).

In summary, in terms of structure the following are all aligned with level 1 onwards: self-directed work teams, a management steering committee for SPI, process owners and process improvement teams. Centres of excellence, team building and team-based practices from the P-CMM, and a SEPG, are added at level 3.

2.3.2.2 People

Adler et al (1992) state that no "technical function can aspire to excellence without policies developing and capitalising on people's potential". They emphasise the establishment of appropriate recruitment, development and reward systems for the organisation.

In organisations with immature recruitment practices the recruitment of staff is typically passive and reactive. These organisations generally do no staff planning and recruit on an ad hoc basis as openings develop. Organisations with mature recruitment practices tend to plan ahead for personnel needs and base these personnel plans on known competency and capability development needs. (Adler et al, 1992) This is consistent with the requirements of attaining a level 3 maturity in the P-CMM, which includes Knowledge and Skills Analysis, Workforce Planning, Competency Development and Competency-Based Practices KPAs (Curtis et al, 1995).

Where the organisation needs staff with the skills and competencies that support its defined standard processes, recruitment practices which are rooted in workforce planning based on an understanding of the organisation's long-term competency requirements, strongly support the attainment of level 3 software process maturity or capability, . The SPICE model's *Provide Skilled Human Resource* process, which is specified as supporting the attainment of level 3 capability, also contains practices that support recruitment and training based on the needs of the organisation (ISO, 1996b; ISO, 1996e). This further supports the allocation of this sort of recruitment practice at level 3.

In addition to recruiting staff with the required skills, an organisation must also develop and train its people appropriately. Training in SPI, an SPI model and quality management principles is required at all levels (Harkness et al, 1997; Risks in Software Process Improvement Programmes, 1997). Training specifically targeted at developing competencies required by the organisation's standard processes is a requirement of level 3 in both the SW-CMM and SPICE models (ISO, 1996e; SEI, 1995).

Competency-based training and development is but one approach to learning that an organisation may adopt as its software processes mature over time. It is only really possible to perform once the organisation has defined its standard processes at level 3. This does not preclude the use of other forms of learning at all maturity levels. A number of learning approaches have been discussed previously in section 2.2.3.1.

Whatever approaches to learning an organisation adopts, it should have a learning strategy in place in order to help it achieve the required levels of learning. Figure 2-7 presents an example of such a model of learning strategies that recognises that an organisation's learning strategy must change together with its environment (Baldwin, Danielson & Wiggenhorn, 1997). The stages in the model are not independent or exclusive and organisations may well have elements of all three stages at any point in time (Baldwin et al, 1997). The usefulness of models such as this one for organisations embarking on SPI is that they provide insight into the learning practices an organisation should adopt to support its SPI objectives and circumstances.

For example, it is arguable that Baldwin et al's (1997) Stage I learning strategy is appropriate where processes are performed on an ad hoc basis with differences of approach between projects. Here success is largely dependant on the skills and abilities of individuals. This would correspond to levels 1 and 2 of process maturity or capability.

Stage II learning strategies are appropriate at levels 3 and 4. At these levels the organisation defines common processes based on best practices and organisational goals (Level 3), and it emphasises gaining control over these processes in order to ensure that the objectives of the processes are met (Level 4).

Stage III learning strategies would support continual improvement and innovation in the face of an unknown or changing environment. This is consistent with Level 5's focus.

<u>Stage I</u>	<u>Stage II</u>	<u>Stage III</u>
Employee Development	Imminent Business Needs	Unknown Business Development
Scope: Individual skill/knowledge Enhancement in current business practices	Scope: Innovation in current business practices to achieve strategic objectives	Scope: Business Redefinition to lead industry restructuring
Focus: Internally defined systems, procedures, and perspectives	Focus: Customer defined requirements	Focus: Undefined market potential
Environmental Turbulence: Low to moderate	Environmental Turbulence: Moderate to high	Environmental Turbulence: High to very high

Figure 2-7: Model of Learning Strategies (Baldwin et al, 1997)

Reward systems and performance management can have a significant impact on the success of an SPI initiative. For example, if a software organisation's staff are only rewarded on the basis of "delivery at all costs", then SPI will at best be seen as a peripheral activity and will not attract the attention necessary to its success (Larner & Bray, 1997). It is therefore imperative that appropriate behaviour at each maturity or capability level is encouraged through the definition of appropriate measures of success at each level.

Although the indicators of success in a software organisation are not only determined by SPI considerations, a number are proposed from an SPI perspective. At all levels a focus on processes, process improvement, quality and customer satisfaction should be rewarded. To support the attainment of specific levels, new measures of success should also be added.

For example, at level 2 meeting project schedules and delivering quality software must be emphasised. At level 3 the recognition of the use of tailored versions of the organisation's standard processes should be added, and at level 4 staff should be rewarded for meeting the process and quality targets set for the organisation. At level 5, rewards for proactive innovation and improvement to better satisfy the customer must be added to the set of performance measurement criteria.

2.3.2.3 Information and Technology

In the context of an SPI initiative there are two classes of tools that should be considered. Firstly, there are those that automate the software processes, for example CASE tools, project management tools, testing tools, defect tracking tools and configuration management tools. Secondly, there are the tools that directly support the practice of software process management and improvement, for example, process documentation and simulation tools.

a. Tools that automate the software processes

Despite great promise and expectations, the use of technology to automate software development and maintenance processes has had mixed results. Some research has pointed to gains in productivity and quality, albeit usually of an incremental nature only, whereas other research has found the expected benefits of automation in the software development and maintenance contexts to be elusive (Orlikowski, 1993; Guinan, Coopridner & Sawyer, 1997).

The work of Guinan et al (1997) and Downes et al (1996) may point to a reason for this. Guinan et al (1997) performed a multi-year study on the use of automated tools in application development on 100 projects in 15 different companies. One of their conclusions is that for teams with well-structured processes, the use of automated tools enhanced the process and improved performance. Conversely, teams not using a well-defined methodology together with CASE tools were significantly less efficient and did not exhibit meaningful improvements in development performance.

Downes et al (1996) concluded from their study of multiple European Community sponsored software best practice improvement projects that "to be effective the introduction of new technologies into software development must be accompanied by (if not preceded by) the implementation of related software development processes". They go on to say that once the processes are in place, the introduction of automated support tends to reinforce and improve the performance of the processes. Jones (1996) supports these findings, stating that heavy investments in tools should be made after the processes are implemented, since the tools support the processes.

The frequently disappointing results of applying tools to software development and maintenance activities could perhaps be due to the generally low software process maturity of the software industry. For example, only about 15% of organisations assessed according to the SW-CMM model are deemed to be level 3 or higher organisations (SEI, 1997). It may therefore follow that the successful, widespread use of tools for the development and maintenance of software may be strongly influenced by the presence of mature software processes in the organisation.

From the above one can conclude that the use of automated tools to support a process cannot be expected to add real value until the process is well defined and repeatable. This occurs at level 3, where the organisation's processes are defined and standardised, taking current best practice and the organisation's goals into consideration. A possible exception is those processes that directly support the SW-CMM level 2 process, for example project planning or software configuration management; or the process areas that directly contribute to the attainment of SPICE level 2 capability, namely those in the Management or Support process categories (ISO, 1996b). Here automation can be used to support established, stable activities (SEI, 1995).

Therefore, the large-scale investment in CASE for software development or maintenance activities is regarded as a change agent that is complementary to a level 3 software process maturity or capability. Some of the supporting processes can be judiciously automated at level 2. However, the tools used for these processes may change at level 3 where I-CASE and Integrated Project Support Environments (IPSEs), consisting of integrated engineering, support and management tools (Sharon & Bell, 1995), may better support the integrated performance of the processes (for example, the Integrated Software Management KPA in the SW-CMM). The use of CASE tools before level 3 can, at best, be expected to support isolated projects and deliver wide variations in performance and in value added.

b. Tools that support the practice of software process management and improvement

The use of tools to support SPI can take many forms. Examples mentioned previously include metrics collection tools and repositories (Haley, 1996; Diaz & Sligo, 1997), software process management tools (Process Management White Paper, 1997), communications technologies such as groupware (Guha et al, 1997; Haley, 1996), and the Internet or Intranets to publish information (Wigle & Yamamura, 1997).

A process management tool is used to help an organisation capture, define and distribute its processes. These tools allow an organisation to maintain its standard software processes in a process repository or library, to tailor these processes to the needs of a specific project, to develop a project work breakdown structure and schedule, and to track progress during project execution (Budlong, Szulewski & Ganska, 1996).

Information on the process's roles and responsibilities, measurements for estimation and process improvement, methods, deliverables, process flow, techniques and tools, are defined in the process management tool's library. The distribution part of the tool has two aspects: the ability to generate project customised plans based on the processes, and the ability to provide developers with access to the process tool's library so that they can benefit from on-line, hypertext or Intranet-based descriptions of the process. The execution aspect of a process management tool entails workflow-type facilities to distribute project tasks and capture feedback about progress from project teams. This feedback, and other feedback from practitioners, is then used to make improvements to the software processes which are captured in the tool's library for future use. (Process Management White Paper, 1997)

Software process management, and other process publishing technologies, will be most valuable in a context where the organisation has standardised processes in place. Once again this occurs at level 3. At lower levels the processes vary greatly between different parts of the organisation and projects, making the publication of standard process definitions and measurements impractical.

Another area of automation that can support SPI is the use of process simulation tools. Dynamic simulation, used in conjunction with systems thinking, process modelling and metrics techniques can be used to create a "microworld". That is, an interactive computerised environment that simulates a real-world situation. A microworld can be used to explore assumptions underlying process definitions and the impact of changes on the processes, or the introduction of new technologies before they are made (Rubin, Johnson & Yourdon, 1995). Because process simulation requires the availability of quantitative data, the use of process simulation tools is only likely to be viable in level 4 and 5 organisations (Burke, 1997).

2.3.2.4 Management

Management's role in an SPI initiative is to provide the leadership and resources required to address the environmental, strategic, structural and procedural factors in all the change dimensions that are required to facilitate and sustain the initiative. Management's style, systems and practices must therefore be orientated towards the achievement of SPI.

Issues such as management's strategic focus and orientation, the use of reward systems to encourage SPI, the allocation of the required improvement resources, the use of measurements and assessments to drive and monitor SPI, and the establishment and management of structures, policies and procedures that support the required behaviours have all been addressed previously. These management practices are appropriate at all levels of software process maturity or capability, although the detail may change at different levels.

With regards to measurements, two other aspects are worthy of elaboration. The use of measures to gather information that facilitates learning and improvement, and the use of benchmarks to compare performance to that of other organisations and to help set process performance and improvement targets.

Davenport and Beers (1995), in research on the use of information in the management and improvement of business processes, identify two types of process information and then relate these to "double-loop learning" ideas. One loop can be described as feedback between a narrowly defined task and its outcome. For example, a worker in a process that refines his or her work procedures to become more efficient is applying single-loop learning by focusing on the effects of his or her behaviour on the process and its outputs. For process information, this loop can be thought of as a *performance* loop, describing process and/or process output performance. See figure 2-8.

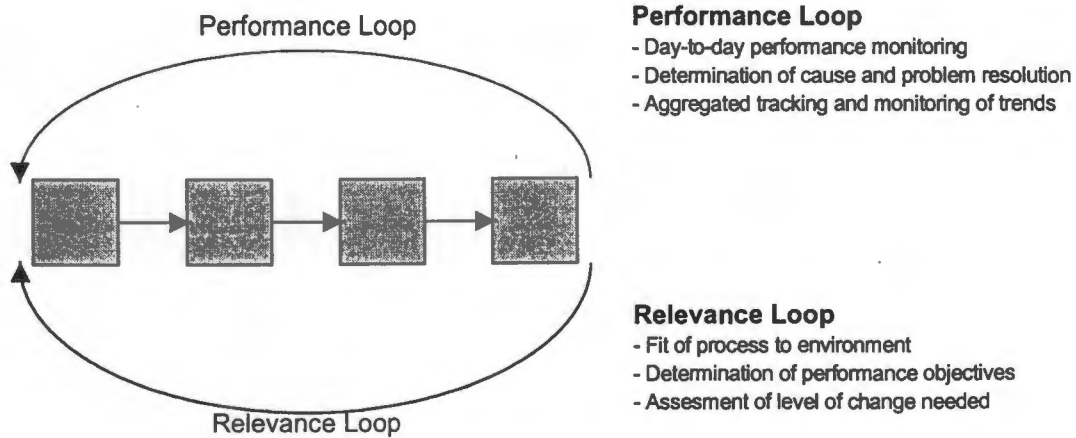


Figure 2-8: Two learning loops for process information (Davenport & Beers, 1995)

The other loop of double-loop learning requires an evaluation and judgement of the broader goals of the task or process. If that same worker stepped back to question whether or not his or her particular task was necessary or appropriate or might be best employed at another stage of the process, he or she would be practising double-loop learning. Davenport and Beers (1995) term this second loop a *relevance* loop, generating information that describes the relevance of the process to the environment. Double-loop learning therefore requires not only a focus on task efficiency, but also on the evaluation of the nature of the task itself and how it fits into a more broadly defined process.

Relevance loop information can be used to help answer questions regarding the processes, such as: (Davenport & Beers, 1995)

- What is the real purpose of the process?
- How does it support our strategy?
- To what degree does it meet the real needs of our customers?
- What performance objectives should it achieve?
- In an ideal world, how should this process be performed?
- What aspects of the process should be completely eliminated?

Both performance loop and relevance loop information are needed for SPI. Performance loop information is useful at all software process levels since it can be used to monitor the performance of any process instance at any maturity level. Even ad hoc processes can be observed and adjusted, albeit in a very context specific and unrepeatable manner. The use of performance information to identify schedule and quality problems begins to be formalised at level 2. At level 3 it can be used for aggregated tracking and trend monitoring since all projects use a process customised from the organisation's standardised processes. At level 4 it is used to quantifiably stabilise the process further through process instrumentation and control and at level 5 performance information can be put to use for defect prevention.

Relevance loop information helps align the processes with the organisation's strategies, performance goals and context. This sort of information therefore starts becoming more relevant at level 3, where the organisation defines its standard processes and their goals to, *inter alia*, contribute to the business goals. At level 5, relevance information is required to guide decisions about appropriate process targets and related improvements, process innovations and the introduction of new technology.

One of the ways of gathering relevance loop information is to use benchmarks of process performance in other firms (Davenport & Beers, 1995). Benchmarks of the software process performance of leading software organisations or the IS organisations of competitors are also useful indicators of process performance targets and SPI objectives that may be relevant to a particular organisation. They also serve the purpose of helping to set strategic performance targets and keeping management interested in the on-going improvement.

Although using benchmarks could provide an indication of target process performance at lower maturity levels, they are only really reliable management pointers at level 3, at which point it is possible to compare an organisation's performance with that of other organisations. This follows from the fact that it is difficult to gather organisation-wide measures for comparative purposes before the organisation has consistent processes which are used across all projects. Before this occurs the variation between projects reduces the reliability of aggregated measures and targets. Conversely, in order for benchmarks gathered from other organisations to be useful, they have to be based on consistent process performance.

As an aside, assessments based on the CMM and SPICE can be seen as a form of benchmark of process maturity and capability. These "benchmarks" are obviously applicable at all maturity levels.

2.3.2.5 Other Enablers

Reuse is often touted as an important approach to improving the performance of a software organisation. The benefits of reuse as a means of reducing time-to-market and risk, whilst at the same time increasing productivity, quality and flexibility, are well documented (Henry & Faller, 1995; Lim, 1994; Short, 1997).

The importance of reuse in the context of improving software process capability has also been recognised. For example, Shea (1997) describes an effort at Boeing to facilitate reuse through increasing software process maturity and using domain engineering approaches. Ragan and Reifer (1998) explore extending the Software Acquisition Capability Maturity Model, a sister framework to the SW-CMM, with product lines, architectures and software reuse. This is consistent with the work of the Software Engineering Institute on version 2 of the SW-CMM, which reflects reuse considerations in the draft of this new version of the SW-CMM (Ragan & Reifer, 1998; SW-CMM v2 Draft C, 1997).

The European Software Institute, a multi-national, non-profit, membership-based organisation, supported by the European Community sponsors a number of projects on SPI, including one that addresses integrating the SW-CMM, SPICE and a reuse capability model developed by another project in this programme (Uniframe, 1998). Jones (1996) also includes reuse as an aspect of his software process improvement model.

Successful systematic software reuse across an organisation requires that a number of elements be in place. Areas that have been identified include a common software process and standards which support reuse practices, reuse education, high quality reusable assets, reuse measurement, a quality assurance programme, configuration management, and a well articulated and communicated business strategy so that longer term development needs and product lines can be targeted (Frakes & Fox, 1995; Downes et al, 1996; Hunter, 1996; Shea, 1997; Ragan & Reifer, 1998). Ad hoc reuse, or reuse within a single project or development team, is the best that can be planned for if these elements are not present in an organisation.

Most areas required for systematic software reuse are addressed by level 2 and 3 software process maturity levels. For example, quality assurance and configuration management will be stabilised at level 2. A common software process and standards driven by business goals, the development of people according to these processes and standards, and the ability to measure performance across projects, are characteristics of a level 3 organisation.

Systematic software reuse across an organisation is therefore only likely to be achievable at level 3 and above. This is consistent with the SPICE model, which addresses some reuse considerations in its "Provide Software Engineering Infrastructure" process. This process mostly supports the attainment of level 3 maturity (ISO, 1996b; ISO, 1996e). The draft of the SW-CMM version 2 includes a new KPA, "Organisation Asset Alignment" which is aligned with level 4 maturity and addresses reuse (SW-CMM v2 Draft C, 1997). This once again confirms the necessity for a defined process, etc. for reuse to be sustainable.

Successful reuse is also strongly supported by the presence of an enterprise architecture in an organisation. The converse is also true (Hunter, 1996). Cecere (1998) describes an enterprise architecture as "a far reaching concept that comprises the vision, principles and standards that govern the deployment of technology". It describes the hardware and software capabilities of the organisation, the treatment of data (what is to be standardised, where it is to be located, and so on) and the treatment of applications (Rockart et al, 1996).

An enterprise architecture enables the alignment with business goals, consistent processes and best practices for software reuse (Cecere, 1998). Other benefits of enterprise architectures include increased flexibility, better integration between information systems, more consistent technology acquisitions (Magrassi, 1995), long term IS planning (Hunter, 1996), increased stability and resilience, and a reduction in cost due to reduced duplication and increased co-ordination (Stevenson, 1995). Cross et al (1997), Ross et al (1996) and Rockart et al (1996) all extol the benefits of architectures in enabling an IS organisation to perform.

Enterprise architectures are also effective in helping an organisation achieve level 3 software process maturity (Cecere, 1998). This follows from the fact that enterprise architectures a) help define the IS organisation's dealings with technology, data and applications in terms of the organisation's business goals, b) imply certain standard processes, including software processes, that apply across the entire organisation, and c) help ensure consistent application of the standards and processes. At level 5 architectures will help guide the introduction of new technology and the ordered change of the organisation's software processes.

Adler (1992) describes how many successful companies have used external relationships as a means to acquire and exploit new technological capabilities. Examples of the approaches used by these companies include acquisitions, mergers, strategic alliances, industry consortia, and joint developments with universities or government laboratories. Exploiting these sorts of relationships or working with organisations that have more mature and capable software processes is one way of strengthening current process capabilities and acquiring new ones. Rockart et al (1996), for example, describe how some companies set out to work with companies with desired capabilities in an effort both to build systems and educate their people.

The exploitation of appropriate relationships with organisations that have more mature software processes is therefore a way of facilitating the SPI and increasing one's own software process capability. These sorts of relationships will arguably assist the practice of SPI at all maturity levels.

In summary, the discussion on SPI and complementary change dimensions highlights the value of enhancing the use of a well-defined SPI model by paying careful attention to complementary "levers" that should be addressed to best support the process improvement and management objectives at each of the software process maturity or capability levels.

2.3.3 Change Management

SPI typically requires a significant change in the software organisation's culture, the ways in which people view their work outcomes and the way that people perform their tasks. This sort of change invariably leads to resistance, both by individuals and groups. If not adequately addressed this resistance can result in the failure of the SPI initiative. Change management is therefore concerned with putting strategies and actions in place that anticipate and overcome possible causes of resistance to change (Randall & McGuire, 1997).

Change management techniques that are often adopted to manage the resistance to change and, equally importantly, to sustain the change within an organisation include:

- Establishing effective two-way communication between the change agents, management and the targets of change. An important element of successful communication is the continual provision of information on the change plans in advance, and timely and honest feedback on the outcomes to all affected parties (Janzon et al, 1997).

- Articulating and monitoring the benefits of the change (Randall & McGuire, 1997) is closely related to the above.
- Involving the people affected by the change in the planning process and establishing an environment within which it is safe to surface resistance. This has the effect of obtaining increased buy-in from the targets of change, and enabling the inevitable resistance to change to become visible so that it can be dealt with in a positive manner (Myers, 1997).
- Ensuring that the changed processes and practices have the following four characteristics: *benefits* for the individual in that it makes their work easier, less frustrating or more productive, *clarity* in that the new processes and practices must be easy to understand, learn and use, *accessibility* in terms of the availability of process artefacts, information and additional support when needed, and *wholeness* in that the support package provides everything that the users of the processes need (e.g. policies, training, reference materials, procedures, integration with other processes, job aids, tools, etc.). (Myers, 1997)
- Establishing high levels of visible management sponsorship and leadership in the change initiative. Although the visible sponsorship of top management is critical, it is also important to filter the sponsorship and leadership of an SPI initiative down to all levels of management (ISO, 1996g; Janzon et al, 1997). The use of cascading sponsorship has the effect of getting all levels of management aligned behind the change initiative as well as ensuring their commitment to and support of the initiative.
- Providing education and coaching that helps introduce and sustain change by equipping people with the knowledge and understanding required to enable them to buy-in to the required changes and to change the way they behave and work. Hutchings et al (1993), for example, advocate an approach that integrates training and consulting and is targeted at teams or groups, rather than individuals, as an highly effective way of bringing about the changes required to increase an organisation's software capability.

- Using a recognition and reward system, consistent with the effort needed to achieve improvement goals, to encourage the attitudes and behaviour necessary for successful change to occur (ISO, 1996g).
- Emphasising a coaching-and-facilitating management style, over a command-and-control style, when dealing with employees undergoing change (Randall & McGuire, 1997).

It is useful to frame the above sort of change management strategies and actions in a model. Many such models exist. For example Kotter (1995) proposes eight steps to organisational transformation support that can be applied in the context of a process improvement effort. Figure 2-9 depicts this model.

SPI processes, such as the Software Engineering Institute's IDEAL approach to SPI (Gremba & Myers, 1997) or the SPICE SPI steps (ISO, 1996g) described previously, address a few limited change management aspects. They do not, however, explicitly deal with all the change management issues typically required to realise large-scale change effectively. It will therefore be useful to integrate and align these SPI processes, which guide the execution of an SPI project, with a change management model such as that proposed by Kotter (1995).

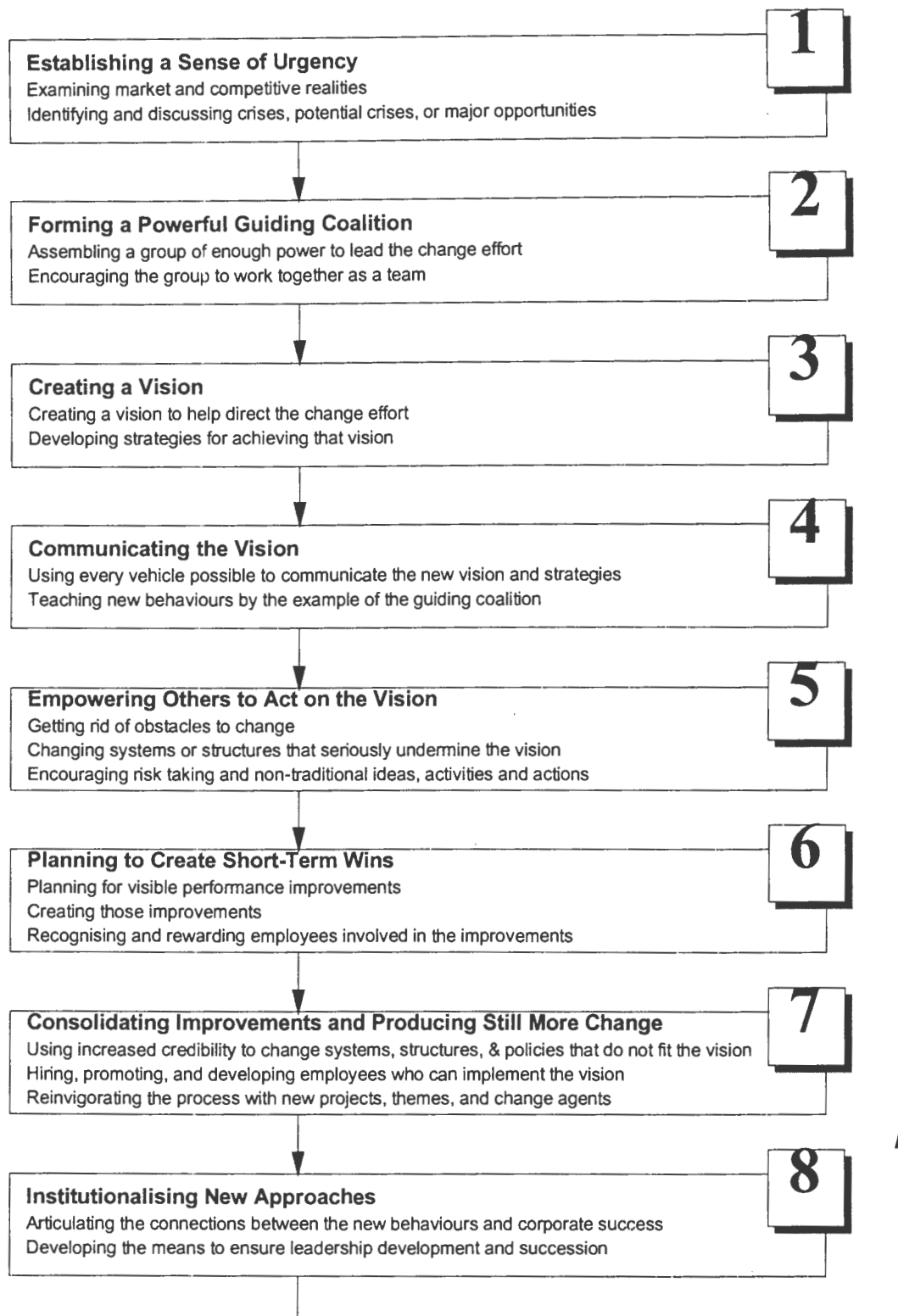


Figure 2-9: Eight steps to organisational transformation (Kotter, 1995)

Most traditional models of change management assume that all changes can be anticipated and planned for in advance. Orlikowski and Hofman (1997) argue that this traditional way of thinking about change is inconsistent with the way that organisations actually implement change, especially in flexible and uncertain organisational and environmental conditions. They therefore propose an *improvisational model* for managing change. This change model rests on two major assumptions that differentiate it from traditional models of change. Firstly, the sort of changes that it applies to constitutes an on-going process, rather than an event with an end point, after which the organisation can expect to return to a steady state. Secondly, most technological and organisational changes made cannot, by definition, be anticipated ahead of time.

Given these assumptions, the Orlikowski and Hofman (1997) improvisational change model recognises three different types of change: anticipated, emergent and opportunity-based change. Anticipated changes are planned ahead of time and occur as intended. Emergent changes arise spontaneously from local innovation and are not originally anticipated or intended. The third type of change, opportunity-based change is not anticipated ahead of time, but is introduced purposely and intentionally during the change process in response to an unexpected opportunity, event or breakdown.

These three types of change build on each other iteratively over time. While there is no predefined sequence in which the different types of change occur, the change process usually begins with an initial anticipated change, followed by a series of opportunity-based, emergent and further anticipated changes. The order of the changes cannot be determined in advance because the changes interact with each other in response to outcomes, events, and conditions arising through experimentation and use.

Considering the depth and breadth of change required by an SPI initiative, the long-term nature of SW-CMM or SPICE-guided change, the many aspects of software practice and the organisation that it impacts, and the rapidly changing business and technological environments within which SPI typically occurs, Orlikowski and Hofman's (1995) improvisational change model seems highly appropriate. Using this change model to manage the change associated with an SPI initiative requires a set of processes and mechanisms to recognise the different types of change as they occur, and to respond to them effectively (Orlikowski & Hofman, 1997). Finally, Orlikowski and Hofman's (1997) improvisational change model is compatible with Kotter's (1995) change model: The fact that Kotter's (1995) change model makes allowances for repeated change cycles makes it possible to accommodate all three change types from the improvisational change model within the greater context of the Kotter (1995) model.

In summary, successful change management creates a climate conducive to change and ensures that the affected change is institutionalised in the organisation. Change management helps overcome resistance to change by ensuring that changes are understood, seen to be beneficial, sponsored by all management levels, open to widespread and active participation, and supported by appropriate education, communication, management, recognition and monitoring strategies. It is useful to place these change management practices in the context of a change management process that addresses anticipated, emergent and opportunity-based types of change and which is integrated and aligned with the SPI process.

2.4 Change Outcomes

The final component of the SPIF is concerned with the results of the change implementation. In the context of SPI, the change outcomes can be classified into three types: Increased process maturity and capability, improved process outputs, and an improvement in the quality of work life (QWL) of employees. These outcomes have all been discussed previously and are only briefly addressed here.

Increased process maturity and capability indicates the extent to which an organisation's processes are stable and improvable. The benefits include having more predictable processes with known ranges of performance and risk, processes that are stable enough to introduce improvements in a controlled manner, and, to a growing extent, increased credibility in the software community.

Improved process outputs are typically measured in terms of improved time-to-market, productivity, quality and customer satisfaction. Hunter (1995) astutely points out that process capability is value-free and that it does not provide any indication of the intrinsic value of the process from the point of view of the organisation that owns it. A process can, for example, be extremely capable in ways that have little or no value to the organisation. A process, however, is capable *and valuable* when it is able to consistently produce results that are aligned with the business goals of the organisation. The improved process outputs should therefore be indicative of added business value to the organisation.

Increased quality of worklife is realised in improved working conditions, increased job satisfaction and pride in work, and strengthened commitment to the organisation and its customers (Kettinger & Grover, 1995).

Generally one would expect all three types of outcome to improve over time. This seems to be the typical case in the SPI literature. However, at least one exception has been recorded: Diaz and Sligo (1997) noted that although quality, productivity and cycle time improved overall during an SW-CMM SPI initiative at Motorola's Government Electronics Division, a decrease in productivity and an increase in cycle time was observed as the organisation moved from level 2 to level 3 maturity. Diaz and Sligo (1997) ascribe this unexpected outcome to the larger change demands expected from development staff as they move from SW-CMM level 2 to level 3.

2.5 Literature Study Summary

Tables 2-9 and 2-10 summarise the findings described in this literature study. These tables represent an Extended SPI Model (ESPIM) that is based on the structure of the SPIF. The first three objectives of the research, described in section 1.5, are therefore satisfied through the literature study in that 1) complementary improvement considerations that the SW-CMM and SPICE SPI models do not address in detail have been identified; 2) the improvement considerations are related to the SPI models; and 3) an ESPIM was defined.

The first table provides an overview of the ESPIM. Table 2-10 provides a more detailed view of a part of the ESPIM and shows how non-process types of change can be aligned with the various maturity or capability levels. It should be noted that table 2-10 follows the approach adopted by the SW-CMM model in that each complementary improvement area is assigned to only one maturity level and is then deemed to apply to subsequent levels.

Framework Component	SPI Elements or Considerations
Change Environment	
Strategic Initiatives	<p>Driven by a clear business need and aligned with a business strategy.</p> <p>Measurable objectives are set.</p> <p>Top management leadership and sponsorship to SPI.</p> <p>Organisation-wide buy-in and commitment.</p>
Change Resources	<p>SPI costs and benefits have been determined and are understood.</p> <p>Long-term commitment to improvement resources and funds is obtained.</p>
Change Facilitators	<p>A culture and management orientation inclined toward change and innovation.</p> <p>A learning capacity in the organisation.</p> <p>A knowledge sharing capability.</p> <p>Higher levels of learning-related scale, related knowledge and diversity.</p>
Change Context	<p>Elements of an SPI friendly change context include:</p> <ul style="list-style-type: none"> • Sound business imperatives to embark on SPI. • Corporate strategies that support the use of IS for business advantage. • The presence of complementary corporate improvement initiatives. • A culture that facilitates continual improvement and innovation. • The resources and size to carry an SPI initiative. • A dependance on software. • Appropriate IS resources, organisational structures, policies and practices. • IS staff with the required skills and an inclination towards SPI.
Change Implementation	
Software Processes	<p>SW-CMM or SPICE Model-based software process improvement</p> <p>A software process improvement process</p>
Non-Process Change Dimensions	<p>Complementary changes can be made in the following areas:</p> <ul style="list-style-type: none"> • Structure • People • Information and Technology • Management • Systematic Software Reuse • Enterprise Architectures • External Relationships
Change Management	<p>Strategies that overcome resistance to change and then sustain the change.</p> <p>A change management process that addresses anticipated, emergent and opportunity-based types of change and is integrated with the SPI process.</p>
Change Outcomes	<p>Increased process maturity and capability.</p> <p>Improved process outcomes.</p> <p>Improved Quality of Worklife.</p>

Table 2-9: The Extended SPI Model

Maturity or Capability Levels	CMM Key Process Areas	SPICE Process Attributes	Non-Process Change Dimensions			
			Structure	People	Information and Technology	Management
1 Ad hoc or Performed informally	<ul style="list-style-type: none">• None	<ul style="list-style-type: none">• Process Performance	<ul style="list-style-type: none">• SDWTs• Management Steering Committee for SPI• Process Owners• Process Improvement Teams	<ul style="list-style-type: none">• Baldwin et al's (1997) Stage I learning strategies• Provide training in SPI and quality• Reward focus on process, SPI and quality	<ul style="list-style-type: none">• SPI friendly management orientation, strategy, structures, systems and practices• Use of performance loop information	<ul style="list-style-type: none">• Exploitation of external alliances with more mature organisations
2 Repeatable or Managed	<ul style="list-style-type: none">• Requirements Management• Software Project Planning• Software Project Tracking & Oversight• Software Subcontract Management• Software Quality Assurance• Software Configuration Management	<ul style="list-style-type: none">• Performance Management• Work Product Management		<ul style="list-style-type: none">• Reward meeting project schedule & quality plans	<ul style="list-style-type: none">• Automated tools that support SW-CMM level 2 KPAs or the SPICE Management and Support Process Categories	

3 Defined or Established	<ul style="list-style-type: none"> • Organisation Process Focus • Organisation Process Definition • Training Programme • Integrated Software Management • Software Product Engineering • Intergroup Co-ordination • Peer Reviews 	<ul style="list-style-type: none"> • Process Definition • Process Resource 	<ul style="list-style-type: none"> • Centres of excellence • Team Building KPA (P-CMM) • Team-Based Practices KPA (P-CMM) • Software Engineering Process Group 	<ul style="list-style-type: none"> • Competency-based workforce planning and recruitment • Competency-based training • Baldwin et al's (1997) Stage II learning strategies • Reward use of standard processes 	<ul style="list-style-type: none"> • CASE and I-CASE tools and IPSEs to support all defined processes • Software Process Management tools 	<ul style="list-style-type: none"> • Use of relevance loop information • Use of benchmarks 	<ul style="list-style-type: none"> • Systematic software reuse • Use of Enterprise Architectures
4 Quantitatively Managed or Predictable	<ul style="list-style-type: none"> • Quantitative Process Management • Software Quality Management 	<ul style="list-style-type: none"> • Process Measurement • Process Control 		<ul style="list-style-type: none"> • Reward meeting process & quality targets 	<ul style="list-style-type: none"> • Process Simulation tools 		
5 Optimising	<ul style="list-style-type: none"> • Defect Prevention • Technology Change Management • Process Change Management 	<ul style="list-style-type: none"> • Process Change • Continuous Improvement 		<ul style="list-style-type: none"> • Baldwin et al's (1997) Stage III learning strategies • Reward proactive innovation and improvement 			

Table 2-10: Mappings from the software process maturity or capability levels to the non process change dimensions

The above tables reflect an idealised view of the mapping between the SPI models and other non-process change dimensions. This does not necessarily mean that successful SPI depends on all these factors being present at any particular point in time, but rather that the success of an SPI initiative is likely to be facilitated by the presence of these factors at each maturity or capability level. The intention is that the work done here provide IS organisations with added insight into what considerations can complement their SPI efforts.

3. Methodology

3.1 Introduction

As raised in section 1.5, the core research questions explored by this dissertation are:

- *What should the change environment for SPI look like?*
- *What other change dimensions should be considered in conjunction with the software process to facilitate SPI?*
- *How should these change dimensions be aligned with the software process maturity levels described by the SW-CMM and SPICE models in the sense of being complementary and supportive of the goals of each maturity level?*

A number of research objectives were defined to assist in answering these questions. The first three objectives have been satisfied through the literature study. The were to identify complementary improvement considerations that the SW-CMM and SPICE SPI models do not address in detail, to relate the identified improvement considerations to these SPI models, and to define an Extended SPI Model (ESPIM) based on the structure of the SPI Framework (SPIF) from this work. (The ESPIM is summarised in tables 2-9 and 2-10.)

The fourth objective was to perform a case study that sets out, in a very tentative way, to determine the extent to which the ESPIM reflects the actual practice of SPI initiatives, to identify reasons for deviations, and to refine the model based on the findings of the case study. A case study was used since it provides details on the context, could provoke discussion with examples, helps one to understand the applicability of the ESPIM in a real setting, and potentially highlight barriers to broader methods such as surveys.

This case study classifies its observations in terms of: 1) areas that are consistent with the findings in the literature survey, 2) areas that were identified in the literature, but that are missing or inconsistent with practice, and 3) areas that exist in practice, but were not identified in the literature. These findings are then analysed and changes to the ESPIM are suggested.

The rationale behind the selection of a site for the case study and the methodology used is described in this chapter.

3.2 Site Selection

The research topic addressed by this dissertation covers a large number of different considerations for the improvement of the software development and maintenance performance in an IS organisation. Given the large breadth of this topic, it is a potentially large and time-consuming task to perform a comprehensive case study to test the model developed during the literature study. It was therefore decided, with an understanding of the limitations that this introduces, to limit the research to one case study.

It was also decided to perform the case study at the company in which the researcher is employed. Two considerations lead to this decision. Firstly, the company had a formal SPI programme in place at the time and, secondly, this offered the researcher easy access to in-depth information. A number of SPI-like projects were in progress in the company and the one that was used for the case study was selected based on the following criteria:

- SPI should be a stated objective and formalised initiative (e.g. a project with appropriate resources) in the departments engaged in the SPI project.

- The SPI project must have been in progress for long enough to enable some meaningful observations to be made. A comprehensive SPI project will typically span a number of years and an SPI project that has only run for a few months will, in all likelihood, not provide sufficient findings to test the literature study conclusions satisfactorily.
- The departments involved in the SPI project under study should ideally address both new software development and maintenance. This would ensure that a broader range of issues, typical of an IS organisation, would be raised.
- The SPI project should have been comprehensive in that it addressed a fair number of software engineering processes or practices, not just one or two very specific improvements. This would better cover the elements of the ESPIM and offer a better range of observations.

Out of three candidate SPI projects the one that best fitted the above criteria was selected for the case study.

3.3 Methodology Procedures

The following is a summary of the steps followed in conducting the case study and analysing the findings:

1. Develop the interview schedule.
2. Perform a researcher self-interview to obtain a first pass set of observations, highlight possible improvements to the interview schedule and identify areas requiring focused interviews.
3. Update interview schedule and prepare for the interviews.
4. Conduct focused interviews to obtain additional observations.
5. Verify findings or details with documentary proof where needed.
6. Analyse the findings and update the ESPIM accordingly.

Each of these steps is described in detail below:

Step 1: Develop the interview schedule

The interview schedule contains a number of questions designed to probe each area in the ESPIM, e.g. strategic initiatives, change context, software processes and change outcomes. The questions do not necessarily explore all the areas in depth, as that would require a far more detailed study than is possible given the scope of the dissertation. Rather, they are intended to cover the breadth of the ESPIM.

Many of the questions were specifically designed to test whether the SPI elements or considerations reflected in the ESPIM were present or applied in the case under investigation. Conversely, other questions sought to identify SPI elements or considerations that were present in the case, but were not identified by the literature study. An example of the former type of question is *"Is the SPI programme represented in your IS department's strategic plans?"*. An example of the latter type of question is *"What major changes, not directly related to the SPI initiative, but pertaining to the functioning of your IS department were introduced during the time period of the SPI initiative?"*

Leading questions were avoided and generally an attempt was made to phrase the questions in *"what"*, *"how"* and *"why"* terms so that they could be raised in an open-ended fashion. The request for examples to illustrate or clarify the respondents' answers is regarded as being an implicit part of just about all the questions.

The final interview schedule, after rework (see below), is included in Appendix A.

Step 2: Self-interview performed by the researcher

Much of the data collected for the case study was obtained from the researcher self-interview. The administration of the self-interview was regarded as a reasonable approach given the fact that the researcher had been intimately involved with the SPI initiative in question from its inception in many different roles and at many different levels. For example, the researcher, as the head of the corporate Software Engineering Process Group (SEPG), was a member of both the SPI initiative's steering committee and project team. This meant that the researcher was in a position to reflect meaningfully on the SPI initiative in question. The use of additional interviews and documentary evidence to verify and enhance the researcher's observations was intended to help balance any bias on his part.

In addition to gathering data, the self-interview served two other purposes:

- To limit the need to administer long, broad-brushed interviews with all respondents by identifying focus areas where specific additional information or insights were needed from particular respondents.
- To test the interview schedule so that it could be refined and updated for the specific focused interviews.

During the self-interview, the interview schedule was applied and observations were recorded and then categorised, per ESPIM area, in one of three ways: a) case study observations that are inconsistent with the conclusions of the literature study, b) areas observed that were not identified during the literature study at all, and c) observations from the case study that corroborate the conclusions of the literature study.

In order to test the usefulness of the interview schedule and to identify areas where additional interviews were required, answers to the following questions were sought and recorded during the administration of the self-interview:

- Is this question, or set of questions, well phrased and clear in terms of what is required?
If not, how should it change?
- Does this question, or set of questions, help extract the required information? If not, how should it change?
- Has sufficient information and insight been obtained from this question, or set of questions in the self-interview? If not, who else can answer the question meaningfully?

Generally, questions that addressed factual issues, such as the software processes that were covered by the SPI initiative, were easily answered during the self interview. Questions that sought to understand less concrete factors were identified as good candidates for additional interviews. The types of questions not answered by the self-interview included those addressing issues such as management's motivation for the SPI initiative and their orientation towards this kind of work, or those that sought to understand people's perceptions of the outcomes of the SPI initiative.

In terms of identifying additional respondents, the following criteria were applied: Firstly, the respondents had to have been involved in the SPI initiative for most of its duration and had to have a good, knowledgeable view of the project. Secondly, the respondents, as a group, had to represent a number of different perspectives on the SPI initiative in order to extract as much additional information as possible and to eliminate any bias on the part of a single respondent.

Therefore, respondents from different departments in the company (from both the software development and the business departments), at different levels within these departments (managers and practitioners), and with different roles in the SPI initiative (the SEPG as change agents, the target departments as recipients of change and an organisational development expert as a facilitator of change) were targeted.

Step 3: Update the interview schedule and prepare for the interviews

The interview schedule was updated based on the lessons learnt during the self- interview. Most of the changes required were fairly minor. Two questions were identified as overlapping significantly with other questions and were removed, a few questions were reworded or extended to improve their clarity, others were combined since they were complementary, and a number of new questions were added. These new questions either probed for more specific information, for example, "*What are some of the specific SPI objectives that are rewarded?*", or were designed to enable more of an open-ended discussion of an area in to uncover additional facts and perceptions, for example, "*What business needs motivated the SPI programme?*".

Once the interview schedule had been updated interviews were arranged with the additional respondents that had been identified in step 2. Preparation for the interviews entailed contacting all the potential interviewees, explaining the nature and objectives of the study and informing them of their expected role and the expected duration of the interview. All of the potential interviewees agreed to participate in the study and interviews were scheduled accordingly. Copies of the interview schedule were produced for each interview, highlighting the questions targeted at each respondent.

Step 4: Conduct focused interviews to obtain additional observations

Five additional interviews were held, ranging in duration from 60 to 90 minutes. Each interview began with a reiteration of the nature and objectives of the study. The fact that the study was not an evaluation of the respondents' SPI initiative, but rather a vehicle to exercise and test the research model, was emphasised. Also, the confidentiality of the identities of the respondents and the fact that the data would not be used by the company to evaluate the SPI initiative was assured. This was important since it placed the focus of the interviews on the research and not the respondents' own work. In this way it was hoped that the respondents would feel more inclined to answer the questions openly and objectively.

Each interview focused on the questions targeted at that respondent, but was also open to delving into other issues. Examples were constantly requested as a way of clarifying the responses to questions. All responses were noted during the interviews. These observations were then used to rework and extend the findings of the self-interview.

Table 3-1 lists all the interviewees and their roles in the SPI initiative.

Who	Role in the initiative
Researcher	Head of the SEPG SPI Project Team member SPI Project Steering committee member
SPI Project Manager	Member of the SEPG Manages the SPI project in the department
Software Development Department Manager	IS Department head SPI Project steering committee member
Business Manager	Business department head SPI Project steering committee member
Software Project Manager	Practising software development project manager Project office manager Member of SPI project team
Organisational development expert	Responsible for change management issues on the project Member of the SPI project team

Table 3-1: Case study participants

In terms of organisational relationships, only the SPI project manager reported to the researcher. However, at the time of the interview the project manager was no longer working on the SPI project under study and had already been appraised for this work. The researcher had no organisational authority over the other respondents and was not required to provide inputs regarding their work performance. This, together with the fact that the research results were not to be used by the company to evaluate of the SPI project, helped minimise concerns regarding the influence of organisational relationships on the research.

Although the number of interviews conducted may seem to be low this was compensated for by the researcher's in-depth knowledge of the company and the particular SPI initiative under study which enabled a comprehensive self-interview to be performed. This is borne out by the fact that towards the end of the series of interviews very few new facts were coming to light during the interviews and most observations were merely confirming previous findings. However, it must be said that had the researcher not had this level of knowledge about the particular SPI initiative a great many more, and longer, interviews would have been required.

Step 5: Verify findings or details with documentary proof where needed

Wherever possible documentary evidence was gathered to verify and cross-check the findings. There were no glaring contradictions to the interview observations in the documents, although one or two additional facts that the respondents had failed to mention came to light.

Table 3-2 lists the documents used and the information that was sought from them.

Document	Information Sought
Area Current Process Analysis report	Initial problem areas "Informal" process assessment results
Change Management Survey (SE Reality Check)	Change management plan's elements and the perceptions of SPI project team members in terms of the success and problems of the SPI Project.
Departmental Software Engineering Approach	SEPG's general procedures for implementing Software Engineering in departments
SPI Project Communication Plan	Change management elements.
SPI Project Cycle Completion Reports	Practices actually implemented Training Provided
SPI Project Cycle Plans	Practices planned for implementation in each project cycle
SPI Project Definition Report	Project goals and objectives SPI project organisation SPI implementation approach Project procedures, including change management elements
SPI Project Budget worksheet	Project costs and time commitments required
SPI Project Plan	Overall project plan elements, including change management aspects.
SPICE assessment report	SPICE assessment results
WPM and SPICE Mapping	The extent of the overlap between the Workplace Improvement Programme's elements and SPICE.

Table 3-2: Documentary evidence sought

Step 6: Analyse the findings and update the ESPIM

The findings of the case study that were found to be either valid or contradictory to the expectations of the literature study, were examined and analysed in order to: a) confirm or challenge the validity of the ESPIM, b) identify the extent to which the conclusions of the literature study were borne out by the case study, and c) propose changes to the ESPIM. The results of this analysis were a preliminary understanding of the validity of the literature study conclusions and an updated version of the ESPIM.

The background and findings of the case study are described in chapter 4.

4. Findings

This chapter describes the background and findings of the case study conducted on the SPI project and concludes with an updated version of the Extended SPI Model (ESPIM).

4.1 Background

The company in which the case study was performed is a large South African financial services company with a number of lines of business, including areas such as managed health care and insurance, life assurance, asset management, pension funds and unit trusts. At the time of the case study (1998) most of these lines of business were supported by a centralised IS division consisting of over 800 people. This IS division consisted of shared support and infrastructure departments, and a number of software development departments that were responsible for providing software solutions and support directly to their allocated lines of business.

The software development department (SDD) whose SPI project was the focus of the case study consisted of about 40 people and was responsible for providing software solutions and application support to the marketing and sales division of the company. They developed and maintained stand-alone PC, client/server and mainframe-based applications for this division.

Two of the business departments serviced by the SDD took part in the SPI project. They were responsible for providing sales support applications and management information to the company's marketers, and acted as intermediaries between the software development department and the real end-users in the marketing division of the company. About 20 staff members were directly involved in this work, mostly in the capacity of analysts, testers and deployers of software applications or management information.

The SPI project in question was initiated by the head of the SDD who was responding to a dictate from the CIO that all SDDs adopt formal software engineering practices, increasing pressure from the business departments to improve the SDD's IS performance, and his desire to develop the SDD, in his words, *"into a world class IS organisation that is the preferred IS vendor to the marketing division and whose staff perceive themselves as a professional team"*. As part of the process to initiate an SPI project the SDD head motivated the two business departments described above to take part in the project and included SPI as a part of his department's strategic plans.

The resultant SPI project involved staff from the SDD, the two business departments and the corporate Software Engineering Process Group (SEPG). The SEPG, positioned in one of the IS division's central support departments, was a group that specialised in software process improvement. It was tasked with assisting the various software development departments of the IS division in their efforts to adopt formal software engineering practices and SPI. An organisational development expert from the IS division was also included on the project team to address organisational change management issues.

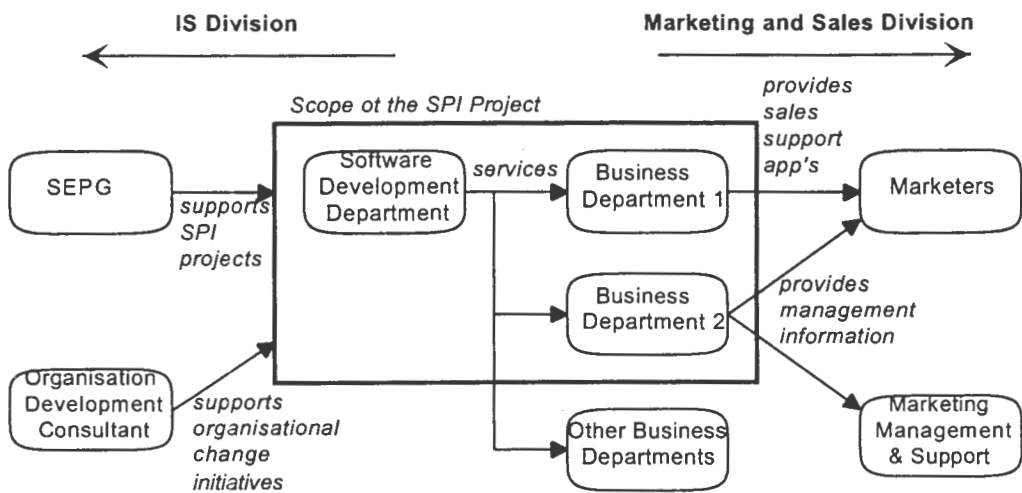


Figure 4-1: The scope of the SPI project described by the case study

The SPI project was originally to run for 20 months, starting in February 1997. Although an SPI initiative would typically run for much longer than this (SEI, 1997), the departments involved were hesitant to commit themselves to a multi-year project before they had some indication of SPI's viability in their context. The expectation was that after the 20-month time frame a decision about a further SPI project would be made based on the outcomes of the first project.

The first activity of the project was an assessment of the software development and maintenance practices of the SDD and the business departments in order to identify potential improvement areas. Some of the techniques applied in this assessment were interviews, current process analysis, and a comparison to best practice as defined by the SW-CMM. Based on the results of the assessment, software engineering processes and practices to be addressed by the SPI project were determined. Examples of problem areas included incomplete requirements analysis, and the fact that there was no single software development life cycle spanning the business departments and SDD.

The software engineering processes and practices selected for implementation were prioritised and allocated to one of a series of four project implementation cycles. The cycles, ranging in duration from 4 to 5 months, were intended to provide a means of focusing the implementation of the targeted software processes and practices into a series of timeboxes that could be more easily managed and tracked. The cycles also limited the number of processes and practices that were addressed for implementation a time.

Within each implementation cycle the SPI project generally implemented the software engineering processes and practices in the following manner:

- The SEPG would define the process or practice generically based on inputs from the literature, the SW-CMM and SPICE models, and previous experience. Training and mentoring requirements would also be recommended.
- The SPI project team and other members of the affected departments, assisted by the SEPG, would customise the generic software engineering process or practice to accommodate department-specific structures and practices.
- Each process or practice was assigned a "jockey" from the SPI project team to co-ordinate its implementation in the software development and business departments. The jockey would work with the SEPG to schedule the training, establish procedures that enabled the use of the process or practice, monitor the implementation, and identify and rectify problems or misunderstandings. Another important aspect of the use of "jockeys" was the transfer of ownership of the process or practice from the SEPG to the departments themselves.

The case study spans the period from the establishment of the project in February 1997 to its premature termination in September 1998. The cause of the SPI project's termination was the disruption caused by a large-scale restructuring of the entire company. This restructuring, which also resulted in the decentralisation of the IS division, resulted in the regrouping of the departments partaking in the SPI initiative, a significant change in the environment of the project, and a change in the senior management role-players that most actively promoted the SPI project. It was therefore decided to stop the SPI project until the whole issue of SPI and its sponsorship and financing could be addressed in the context of the new organisational structures.

Although the SPI project did not run its full course or achieve all of its objectives, all the respondents in the case study expressed their satisfaction with its outcomes at the time of the project's termination, and the fact that they would "do it again". The SPI project also ran for long enough as a concerted project to offer some useful observations for this research, especially in the change environment section of the proposed SPI framework.

4.2 Findings

4.2.1 Summary of the Findings

Table 4-1 summarises the findings from the case study. For each area of the ESPIM the observations are classified in one of three ways: a) case study observations that were inconsistent with the conclusions of the literature study, b) areas observed that were not identified during the literature study at all, and c) observations that corroborate the findings of the literature study. In this way findings were separated into those that challenge the literature study conclusions, those that offer potentially new insights, and those that one would have expected based on the literature.

In addition, a finding is marked with either a plus (+) or a minus sign (-) when it was found to strongly influence, either positively (+) or negatively (-), the execution or overall effectiveness of the SPI project. For example, "Insufficient time was made available for staff to effectively partake in the SPI project" was often mentioned by the respondents as negatively impacting the SPI project, and is therefore prefixed with a minus sign in the findings summary. Note that only those findings that were clearly identified one or more of the respondents as having a positive or negative impact are annotated in this way.

ESPIM Areas		Data and Findings		
		Deviations from the literature study findings (Challenging Findings)	Observations not expected from the literature study findings (New Findings)	Observations that corroborate the literature study findings (Expected Findings)
Change Environment	Strategic Initiatives			<ul style="list-style-type: none"> • SPI was seen to be an important part of the departments' strategy. • + Formal sponsorship involving top management was established. • - Sponsors played a passive role. • General buy-in to the SPI project was initially low which hampered the SPI project.
	Change Resources			<ul style="list-style-type: none"> • Sufficient finances and staff were available for the SPI Project. • - Insufficient time was made available for staff to partake effectively in the SPI project.
	Change Facilitators	<ul style="list-style-type: none"> • The total number of staff impacted by the SPI project was relatively low which would typically make it less likely for the departments involved to initiate or maintain an SPI initiative. This was overcome by the fact that the SPI project was effectively subsidised by the overall IS division. In other words, the learning-related scale of the project could be regarded as falling into the ranges that the literature study would have expected for an SPI initiative to be sustainable. 		<ul style="list-style-type: none"> • The SPI project incorporated a number of approaches to enable and encourage SPI-related learning. • Cultural factors that could <i>hamp</i>er the SPI project included risk aversion, the lack of a shared vision for the project outcomes and a conservative approach to implementing change. • Cultural factors that could <i>enable</i> the SPI project included an openness and desire to change, a management approach that encouraged and trusted staff to make changes. • Initially the SPI-knowledge sharing capability in the departments was low. • Actions to address the low levels of knowledge sharing during the SPI project included defining processes in a process management tool, capturing standards in a Lotus Notes database, creating deliverable templates and establishing a project library. • The levels of related knowledge and the diversity of technical knowledge in the software development department and the SEPG were relatively high and indicative of the likelihood of the SPI project being initiated and sustained. • + The head of the software development department had a developmental orientation which may explain why much of the enthusiasm and drive for

Data and Findings			
ESPIM Areas	Deviations from the literature study findings (Challenging Findings)	Observations not expected from the literature study findings (New Findings)	Observations that corroborate the literature study findings (Expected Findings)
	Change Environment cont.		<p>the SPI project came from him.</p> <ul style="list-style-type: none"> On the whole, most of the other managers in the departments had a tactical or defensive orientation which could help to explain why they often saw the SPI project as a secondary priority.
Change Facilitators cont.			<ul style="list-style-type: none"> The change context was found to <i>support or facilitate</i> the SPI project in the following ways: <ul style="list-style-type: none"> Strong competitive pressures and client demands for the rapid delivery of high quality software contributed to the motivation behind the establishment of the SPI project. The presence of a company-wide improvement initiative supported and complemented the SPI project. The existence of a project office and the "centre of excellence" nature of the organisational structure of the three departments involved in the SPI project made the work of the project a bit easier. + The presence of a strong SEPG was an important enabler of the SPI project. The ability of the organisation to overcome learning barriers made it feasible to embark on the SPI project. The SPI project was <i>hampered</i> by the following elements in the change context: <ul style="list-style-type: none"> The fragmented organisational structures between the three departments involved in the project created gaps in the software processes and resulted in communication, prioritisation and co-ordination problems. - A lack of formal IS policies and associated management behaviours, which might encourage SPI projects in the company, made it hard to motivate and sustain the project. - The massive organisational change in priorities and structures removed the direct influence of key role-players required to sustain the SPI
Change Context			
Change			

ESPIM Areas		Data and Findings		
		Deviations from the literature study findings (Challenging Findings)	Observations not expected from the literature study findings (New Findings)	Observations that corroborate the literature study findings (Expected Findings)
Environment cont.	Change Context cont.			project in the absence of other sustaining forces.
Change Implementation	Software Processes			<ul style="list-style-type: none"> • The Software Process Maturity level of the departments participating in the SPI project could be regarded as corresponding to SW-CMM level 1 with individual processes corresponding to SPICE level 0 and 1. • Although not explicitly following either the SW-CMM or SPICE models, the SPI project followed many of their guidelines. • The SPI project tended to emphasise typical level 2 maturity-type practices for implementation. • The SPI project largely followed an implementation approach that closely resembled the eight SPICE SPI steps.
	Non-process Change Dimensions	<ul style="list-style-type: none"> • + The SEPG played a critical role at level 1 and not just at level 3 as literature indicated. • Characteristics of centres of excellence (especially the grouping of related skill needs) may help implement SPI at levels lower than the expected level 3. • One can start skills-based recruitment and training before level 3 when there are clearly defined roles directly relevant at a lower level. The project manager and analyst roles are examples of this at level 2. • Software process management tools, such as Process Engineer, already 	<ul style="list-style-type: none"> • The presence of a project office at level 1 is very useful in helping to move to level 2-type practices. • The existence of requirements analysis processes is potentially an important enabler of systematic reuse. 	<ul style="list-style-type: none"> • Self-directed work teams and the presence of SPI project roles, such as the management steering committee, process owners and a process improvement team, are likely to assist an organisation engaged in level 1 and 2 SPI activities. • Competency-based staff recruitment and training is only likely to be utilised at higher software process maturity levels than that of the departments covered by the SPI project. • The overall learning strategy of the departments was consistent with that expected of an organisation with a low level of software process maturity. • The lack of SPI-related training may explain the general lack of understanding for the SPI project and the low levels of buy-in in some quarters. • - The lack of SPI-related rewards undermined the effectiveness of the SPI project. • The use of CASE was not effective in the past owing to the lack of an established methodology. • Tools that supported processes targeted at level 2, for example project
Change	Non-process			

ESPIM Areas		Data and Findings		
Implementation cont.	Change Dimensions cont.	Deviations from the literature study findings (Challenging Findings)	Observations not expected from the literature study findings (New Findings)	Observations that corroborate the literature study findings (Expected Findings)
		added value at level 1, and not level 3 as the literature study predicted, in establishing the use of a methodology and software processes		<p>management and software configuration management tools, were more widely used as the departments targeted level 2-type processes.</p> <ul style="list-style-type: none"> • Management did not make use of performance information which made it hard to quantify possible improvements from the SPI project. • Management practices such as sponsoring the initiative, insisting on the use of the new software processes and providing resources for the SPI project were seen to <i>contribute</i> to the SPI project. • - Management behaviours such as not ensuring that staff had sufficient time available for the SPI project or to learn new practices, the lack of rewards to encourage the behaviours required by the SPI project and the fact that managers were sometimes seen to be mere "cheerleaders" of the SPI project lessened the impact that management had on the SPI project's success. • The failure of the attempt to introduce systematic reuse during the SPI project confirms the literature study conclusion that reuse is unlikely to succeed before many of the enabling level 2 and 3 process are in place. • The SPI project included change management strategies that contained a number of elements aimed at overcoming resistance to the required changes. • The SPI project explicitly accommodated both anticipated and unanticipated types of changes.
	Change Management		<ul style="list-style-type: none"> • Formal change management strategies should be complemented with informal, re-inforcing actions, especially on the part of management. • + The capability to respond quickly to unanticipated changes, for example, having a flexible and agile SEPG with access to a range of useful resources, is a key enabler of this sort of change. 	
Change Outcomes			<ul style="list-style-type: none"> • "Soft", or intangible, and quality of work life benefits may be realised in 	<ul style="list-style-type: none"> • The duration of the SPI project was too short to realistically expect any real improvement in software process maturity level or process outcomes.

ESPIM Areas	Data and Findings		
	Deviations from the literature study findings (Challenging Findings)	Observations not expected from the literature study findings (New Findings)	Observations that corroborate the literature study findings (Expected Findings)
		the short term, before increased process maturity or process outcomes become evident.	<ul style="list-style-type: none"> Although improvements in terms of process maturity and outcomes were not measured, a number of other positive outcomes were highlighted by the respondents. Many of these addressed quality of work life or "soft" issues. Many respondents felt that the SPI project actually negatively impacted on their productivity due to longer than expected learning curves.

Table 4-1: Summary of the case study observations

Areas in the literature study which could not be corroborated by the case study are listed in table 4-2. These are the areas of the ESPIM for which insufficient evidence was gathered to arrive at meaningful conclusions.

ESPIM Area	Observation
Change Environment: Strategic Initiatives	<ul style="list-style-type: none"> • The impact of the SPI project was not motivated by a clear business need. • Measurable objectives were not defined.
Change Implementation: Non-Process Change Dimensions	<ul style="list-style-type: none"> • Observations relating to the alignment of practices above level 2 were not possible since the departments were only pursuing level 1 and 2 practices. For example, the alignment of Baldwin's (1997) higher stage learning strategies with higher software process maturity levels could not be tested. • Enterprise architectures were not used. • Although the possibilities appear promising, the external relationships that the departments are engaged in have not yet had sufficient time to impact their software process maturity materially.
Process Outcomes	<ul style="list-style-type: none"> • Expected process outcomes were not formally specified or measured so it is difficult to quantify the possible improvements from the SPI project.

Table 4-2: Areas for which insufficient evidence was gathered to meaningfully comment

4.2.2 Details of Findings

The details of the above findings are discussed in this section. Whenever appropriate, respondents' statements are quoted in italics and examples are given to illustrate the area under discussion.

4.2.2.1 Change Environment

In terms of the change environment, four areas are discussed: the strategic nature of the SPI project, the change resources available to the SPI project, the factors that may have facilitated the changes required to be introduced by the SPI project, and the context within which these changes would occur.

i. Strategic Initiatives

The respondents identified a number of objectives for the SPI project. The most commonly identified objectives were to enable more rapid software solution delivery and to improve the quality of the software that was delivered. Other objectives included improving the software development and maintenance practices of the SDD to incorporate "industry best practices", the creation of a highly professional and skilled group of IS developers, and establishing a single integrated software delivery process, spanning the business departments and the SDD.

However, the above objectives were not clearly defined or articulated in that there was no one agreed list of objectives for the SPI project. It seems that it was more a case of a broad understanding that an SPI project could, in the words of one of the business department managers, make the provision of software solutions "*faster and better*", and there was little effort to formalise the objectives beyond that. For example, achieving a higher software process maturity level or increasing developer productivity by some percentage were never mentioned as objectives of the SPI project.

The fact that the objectives were only loosely defined also meant that they were never stated in measurable terms and that targets for the SPI project, other than for practice implementation, were not set. Also, no formal attempt was made to tie the objectives of the SPI project to business objectives, other than to recognise that the marketing division was moving into a situation where it was releasing new products to the market more frequently, necessitating more rapid application development.

The SPI project was represented in departments' strategic plans despite the fact that its objectives were so loosely defined. The SDD explicitly included the SPI project as one of its two strategic focuses for 1997 and 1998. One business department's strategic plan also included an implicit reference to the SPI project in a strategic goal which aimed to improve the rate at which quality software could be delivered to its end users.

The heads of department of the software development and business departments further took a lead in the SPI project by acting as co-sponsors of the project and serving on the project's steering committee. Although the steering committee adopted a fairly passive role in the project and tended to rely on the project team to set direction, the presence of the heads of the three departments on the committee sent a clear message to all the departments' staff that they regarded the SPI project as being important.

The steering committee also became more vocal and active in its support once the SPI project was seen to be producing benefits and the sponsors began to understand their roles as sponsors more fully. For example, they began to insist that projects apply some of the processes implemented by the SPI project. Unfortunately, this support from the sponsors was not always translated into concrete organisational action, such as changing reward systems to re-inforce and sustain the new software processes and practices. This meant that the sponsors tended to play a "cheerleading" role, but were not always seen to play a tangible part in effecting the changes required by the SPI project.

In order to promote the SPI project as a strategic initiative a series of presentations and workshops with all the staff from the SDD and the business departments were held. These explained the content of the SPI project and its expected benefits. In addition, a number of private discussions were held with individuals to address issues and concerns that they may have had.

Although these communication efforts did help expose the stakeholders to the SPI project, they did not, by and large, manage to create the hoped for levels of motivation for the project. This was ascribed to the possibility that the presentations went over the heads of a number of participants and that many of the practitioners "*could not immediately see the benefits in their own areas of responsibility*". Another barrier to the SPI project was the fact that there was a fair amount of scepticism about this type of initiative from some of the staff. This resulted from a failed effort to implement a new methodology, called Infomet, a few years earlier. Although care was taken to emphasise that the "Infomet mistakes" would not be repeated, many of the SPI project's participants greeted this with a wait-and-see attitude.

In conclusion, there was some understanding of the strategic importance of the SPI project as a means of improving the performance of the departments and better meeting their client's needs. Also, the top managers in the departments attempted to play a visible sponsorship role. However, clearly articulated and measurable objectives that could be explicitly related to a business need were not available, and some of the respondents felt that the sponsors played a more passive role than they ideally could have. Another hampering factor was the relatively low organisation-wide buy-in and commitment to the SPI project in the early stages.

ii. Change Resources

The literature study identified three categories of resources required by an SPI project: budget, staff and their roles, and the time that was allowed for the project.

Initially the SPI project budget was not comprehensive and only the cost of support and participation from the corporate SEPG was explicitly calculated. The costs of the project to the software development and business departments, based on staff and training costs, were regarded as part of the departments' normal annual budgets and were therefore not regarded as new expenditure. What is more, many of the costs incurred by the SEPG were carried as an overhead cost by the IS division and were not factored into the departments' SPI project budget. Examples of these centrally carried costs, which were quite significant, included the purchase of a software process management toolset, the development of generic software engineering processes and practices, and the purchase of a video-based training series in software engineering. The project also did not require the purchase of new tools or the use of external consultants, which kept the costs in check.

Since the SPI project was quite heavily subsidised by the IS division its budget was relatively small. As a result the direct cost of the project to the departments was not regarded by their management as being excessively high.

The SPI project was adequately staffed and a number of different roles and responsibilities were evidenced. These included the following:

- The corporate *SEPG* which managed the SPI project and provided software engineering practice support, guidance, and training.

- The *SPI project team*, which was called the Software Engineering Implementation Group (SEIG): It consisted of representatives from the SDD, the business departments, the SEPG, and an organisational development expert to address organisational change management issues. This group was responsible for the day-to-day project activities, including implementing the software engineering processes and practices. Some individuals in the SEIG were allocated as "jockeys" of certain processes or practices. The role of the "jockeys", as described in section 4.1, was to co-ordinate and facilitate the implementation of the software engineering processes and practices and fulfill the role of process owner for the areas for which he or she was responsible.
- The *project steering committee* which consisted of the three department heads, the manager of the SEPG, and the SPI project manager. The steering committee met regularly to review progress and address scope issues and concerns that the SEIG could not address effectively.
- *Staff* from the software development and business departments were frequently drawn into the project during the implementation of the software engineering processes or practices, training sessions and project feedback sessions. In this way, most of the departments' staff were involved in the project in one way or another, albeit on an ad hoc basis.

The SPI project organisation therefore covered the roles and responsibilities typically expected in a project of this nature. However, one of the most frequently raised complaints was the lack of time allowed to participate effectively in the SPI project.

The SPI project actually people's workloads as they had to take part in the project while maintaining their current workloads. No compensation was made for the additional demands on their time caused by the implementation of new processes, attending training, and overcoming the learning curves associated with the new processes. As a result, the SPI project was frequently criticised for taking too much time from "real" work.

Furthermore, management paid lip service to the possibility that the project might impact negatively on productivity as a result of people working on the SPI project and learning new practices, but this did not translate into the lengthening of project schedules or an increase in project resources. As a result, the SPI project was frequently undermined by "operational pressures" and "business realities" that demanded the rapid delivery of software.

For example, on one project the use of a new methodology and notation for analysis was significantly scaled back when the analyst found that she could not complete the analysis within the aggressive timescales of the project, as they had not been adjusted to allow for the adoption of new practices. In another example, the implementation of a number of software engineering processes and practices was delayed for a few months or, in some cases, removed from the project scope entirely, because the department was not able to make extra time available for its staff to implement and learn these practices.

In conclusion, in terms of the resources required by the SPI project, the areas of project budget and staff to work on the project were well catered for. However, insufficient time was made available for people to effectively participate in the project. This was seen to be a significant hindrance to the success of the SPI project.

iii. Change Facilitators

Based on the findings of the literature study, five different potential change facilitators were explored in the context of the SPI project. These were the learning capacity of the organisation, its cultural readiness for SPI, its knowledge sharing capabilities, its ability to overcome knowledge barriers, and the orientation of its management in terms of SPI.

a. Learning Capacity

An effective SPI initiative requires that the organisation within which it is happening has the capacity to learn and innovate. This was realised by the SPI project in a number of different ways, including the use of training, mentoring, implementation reviews, and the incorporation of industry inputs.

The training took three forms: formal training courses, informal training, and video-based training used to introduce a new process or practice or to supplement the formal training. Formal training included courses in analysis and the use of the software process management toolset, Process Engineer. Informal training was usually presented by a member of the SEPG and took the form of workshops, e.g. on reviews, presentations and one-on-one sessions. Informal training was used where formal training courses on the processes and practices did not exist, or to complement the formal training courses with overviews of company specific standards and practices.

Learning was also facilitated through mentoring, the distribution of useful articles, and discussions within the SEIG. For example, the SPI project manager spent a considerable amount of time with the software project managers, helping them to use the Process Engineer tool and to understand the methodologies that were documented in its process repository.

The SPI project did not use pilot projects to test the new processes and practices prior to widespread use. However, all implementation efforts were monitored in order to identify and resolve problems, either through additional training or by refining the definition of the process or practice. This was one of the responsibilities of the SEIG "jockeys", who were supposed to propose improvements or call in the SEPG expert to help sort out any problems.

Unfortunately this monitoring process did not always happen to the extent that was hoped, largely because many "jockeys" were overloaded with more than one process, and had not had their other responsibilities adjusted accordingly. However, in a few cases the monitoring process did happen as was planned, for example, in the area of project change control. In this case the "jockey" was a project manager who had been mandated by the steering committee to apply change control as a part of everyday project practice. A number of workshops were therefore scheduled to resolve issues that had arisen between the business departments and the SDD during the application of the process. The SEPG expert in project change control was actively involved in these workshops, clarifying areas of misunderstanding and proposing a few changes to the way that the project change control process had been implemented.

Continual learning also took place through formal reviews of the project and of lessons learnt at the end of each project implementation cycle. Issues, problems, success stories, and proposals for changes to the project were highlighted by these reviews and discussed in detail with the steering committee and the SEIG. Presentations of each cycle review were also given to all the staff in the departments involved in the SPI project. In this way everyone shared the benefit of the lessons learnt during the previous SPI project cycle, and future cycle plans were adjusted accordingly.

Finally, new knowledge was introduced during the SPI project through the definition of the generic software engineering processes or practices by the SEPG. These generic processes and practices were based on a number of different sources, including industry models such as the SW-CMM and SPICE, commercially available methodologies, text books, journal articles, other companies' processes and practices, and previous implementations in which the SEPG members had been involved. The generic processes and practices were then adopted by the SEIG and changed slightly, in conjunction with the SEPG, to cater for departmental-specific conditions before implementation. In this way, lessons and "best practice" from a number of different sources were incorporated by the SPI project, while at the same time department-specific needs and conditions were considered.

In summary, the SPI project covered by the case study incorporated a number of approaches to enable and encourage SPI-related learning by the staff of the software development and business departments. Formal and informal training, group and individual learning, feedback and responding to lessons learnt, and the adoption of outside knowledge were all utilised to improve the learning capacity of the staff affected by the SPI project.

b. Cultural Readiness

A number of factors can indicate the cultural readiness of an organisation to undertake a successful SPI initiative. Cultural norms such as risk taking, a shared vision, trust and high expectations for action (O'Rielly, 1989) were, for example, identified as factors facilitating an SPI initiative in the literature study. The case study found mixed indications of a cultural readiness for SPI in the departments involved in the SPI project.

For example, there was little evidence of tolerance towards risk taking and failure. The general feeling was that nothing should be done that could possibly increase the risk that one of the departments' projects might not finish on time. In this sense the SPI project tended to shy away from practices that were seen either to be too different from the status quo or to introduce too much change all at once. For example, the introduction of a methodology incorporating RAD practices was watered down quite significantly by the project managers to avoid many of the less familiar or more "risky" aspects of a RAD project in their planning using the methodology. The SEPG staff on the project also had a sense that they had to tread very lightly since any hint that the SPI project might not perform as expected would be deemed a failure and lead to its immediate termination.

Despite of risk aversion culture and shying away from anything that could be seen as affecting the delivery time of a project, the respondents felt that key and influential staff members were generally open to the required changes and the new ideas introduced by the SPI project. In most cases, there was consensus that adopting "best practice" was important and that they needed to be willing to change. Willingness to change was further facilitated by the fact that a number of staff, particularly those involved in project management and requirements analysis, were quite desperate to learn new and improved ways of fulfilling their responsibilities. What is more, the management of the departments was, on the whole, quite enthusiastic to introduce the changes required by the SPI project, probably since they saw this as a key part of their strategy to improve their departments' performance.

However, despite this general willingness and openness to the changes proposed by the SPI project, many of the practitioners were seen to be quite cautious in the rate at which they were prepared to implement the changes, particularly where they perceived that the changes might possibly influence their work schedules or where they were not initially convinced of the value of the changes. Factors that may have influenced this hesitation included fears that the changes might be another failed "Infomet methodology exercise", the general culture of risk aversion in the company, and a fear of not completing projects in time.

The conflict between the willingness to change and a cautious approach to implementation was often evidenced. For example, the project office head was put under significant pressure by both the SDD and business department management to be more proactive and aggressive than he was comfortable with in the implementation and usage of practices such as reviews, project change control and project planning based on the RAD methodology. This is not to say that the practitioners rejected the change entirely, but rather, that they tended to opt for a more conservative and incremental approach to implementing the required changes.

Although management occasionally mandated the introduction and usage of new software engineering processes and practices, they generally left the definition and implementation of the practices entirely in the hands of the SEIG and the practitioners. In this sense, management clearly trusted their staff and expected them to take ownership and responsibility for making the changes required by the project. For example, the SEIG and the project office determined the exact processes and procedures to be used in all the affected departments for project change control without any additional inputs from the steering committee. The SEIG also had a lot of latitude in setting the content and direction of the SPI project and were able to make decisions about the project scope, approaches and pace.

The last aspect of cultural readiness explored by the case study was the extent to which there was a shared vision for the SPI project. Here it was found that there were widely differing perceptions and expectations of the project amongst the stakeholders, and that there was not a single, clear, shared vision for the outcomes of the project. The SEPG, not unsurprisingly, tended to have a much more optimistic and positive view of the changes that the SPI project would bring about. Some of the senior managers involved in the project, most notably the head of the SDD, also shared this vision to some extent.

The people directly involved in the SPI project on a regular basis, particularly the SEIG, generally had a good understanding of what the project was trying to achieve, but did not always understand the finer details of SPI and software engineering. They also tended to see things in much more operational terms and were, for example, more concerned with answering questions such as "how can change control help me keep control over this project's progress?". They did not, therefore, necessarily share the SEPG's enthusiasm for the "big picture", and were more concerned with realising the day-to-day objectives of the SPI project. Some of them had also decided to reserve judgement on the likely outcomes, given their feelings about the ineffective Infomet methodology implementation mentioned earlier.

Finally, the people "on the floor", so to speak, were quite removed from the overall project vision and were only really impacted by the SPI project when they were directly involved in the introduction of a new set of software engineering processes or practices, or during the occasional project cycle review. In short, people's vision for the change seemed to vary greatly, depending on their level of involvement in the SPI project and their preconceived ideas about SPI.

In summary, cultural factors that hampered the SPI project included a tendency to avoid risk, the lack of a shared vision for the outcomes of the SPI project, and a conservative approach to implementing change - there was generally not a strong bias to action. On the positive side, a culture of openness and desire to change, and a management approach that encouraged and trusted staff to adopt changes in the interests of improvement, were evidenced.

c. Knowledge Sharing Capability (and the use of IT to facilitate this)

The ability of an organisation to capture and share its software engineering knowledge is an important facilitator of an SPI initiative. However, when the SPI project began, the departments involved were typical of organisations with a low software process maturity. Software processes, practices and standards were generally not documented, best practices and lessons learnt tended to reside in individuals' heads, metrics were not collected, analysed or applied for estimation and improvement purposes, and different people continually had to "reinvent the wheel". This situation was aggravated in one team of the SDD by a much higher than normal staff turnover rate over the past three years.

One of the areas that the SPI project, in conjunction with the SEPG, therefore focused on was to capture and record some of this tacit knowledge in a formal way and to make it freely available to anyone that needed it. This was largely realised in four ways:

- Firstly, software development, maintenance and operations standards were formally documented and made available on a Notes Database. A central department in the IS division was also made responsible for promoting the use of the standards and keeping them up to date in conjunction with so-called standards experts - people that know and maintain the standards.

- Secondly, all the defined software engineering processes and practices were documented and made available in the software process management tool's repository. The tool, Process Engineer, was used to provide the software process definitions in a hypertext format to all staff from a central server, making them available to anyone that needed them. For example, the RAD methodology and the project change control procedures were made available in this way. Process Engineer also enables one to generate project plans directly from the documented methodologies. This functionality was used by project managers during project planning, thereby ensuring that the best practices and knowledge defined in the processes were incorporated into projects plans.
- Thirdly, all deliverables required by the software processes were defined and distributed in the form of templates. These templates described the layout and sections of each deliverable, provided explicit guidance on how they should be completed and, in some cases, also incorporated examples. The basic premise behind this was that "a good template implicitly guides the user to follow the process and apply best practices."
- Fourth, a set of procedures for the establishment and management of a project library were defined. Initially this was a paper-based system, but at the time of the case study a Lotus Notes system was being implemented. The benefits of the Lotus Notes system were that it also provided document management, electronic distribution and full text search capabilities on the documents in the database. In this way, project documentation, including issues and lessons learnt, were managed and made more freely available to people who needed to access them.

An area that was not addressed by the SPI project was the establishment of a metrics database to track productivity and quality metrics for estimation and process improvement purposes.

An exception to the lack of defined and documented software processes and practices was observed. One of the business departments involved in the SPI project had previously made an effort to define and document their software procedures in some detail, to apply them in their work, and to train new staff members in their use. Unfortunately, these procedures were not always integrated with the software development and maintenance practices of the SDD that was responsible for performing most of the technical work. It was also found that some of the procedures were incomplete or dated. Having recognised these problems the business department began to update their software procedures to incorporate many of the processes and practices introduced by the SPI project, to integrate them with the SDD's processes, and to document them in Process Engineer.

In conclusion, although the knowledge sharing capability of the three affected departments was low at the outset of the project, much was done to rectify this situation during the timeframes covered by the case study. In addition, IT enablers such as a software process management toolset and a groupware environment, namely Lotus Notes, were extensively used to facilitate the development of this knowledge sharing capability.

d. Overcoming Knowledge Barriers

Research by Fichman and Kemerer (1997) found that factors such as a greater scale of activities over which learning costs can be spread (learning-related scale), more extensive knowledge in areas related to an innovation (related knowledge), and a greater diversity of technical knowledge and activities in general (diversity) affect the propensity of an organisation to assimilate initiatives such as SPI. It was also found that larger organisations tend to have these characteristics.

The three departments involved in the SPI project had a staff complement of about 60 people. The cost of the SPI project, as discussed in the section on Change Resources, was heavily subsidised by the IS division as an overhead cost, and was therefore significantly lower than it would have been if the departments had to carry the entire cost of the project themselves.

Therefore, although the number of staff involved in the project was comparable to that of a relatively small IS organisation, which would typically be less likely to be able to undertake an SPI initiative [Downes et al (1996) draw the line at 100 staff members], this was largely overcome by the fact that much of the cost of obtaining the expertise for the SPI initiative was in effect spread over a much larger organisation, namely the entire IS division of 800 people. If this were not the case the SPI project would probably not have been sustainable for as long as it was because the costs would have been higher than the departments could justify by themselves.

The SPI project was also facilitated by work done in related fields in the past. Specifically, the attempted implementation of the Infomet methodology, the implementation of the IBM MITP project management methodology, and the implementation of an object-oriented methodology and CASE tool a few years back, all resulted in the staff having had exposure to the concepts and practices that the SPI project introduced. In many ways this eased the burden of having to introduce many software engineering processes and practices from first principles. It also meant that, in some cases, the SPI project was a natural continuation of previous efforts and that people had some idea of what they wanted when the implementation of the processes and practices was discussed.

Conversely, some of the previous exercises, particularly the Infomet methodology implementation, had created some negative perceptions of what the SPI project was trying to achieve. These perceptions had to be carefully managed, and the differences between the two exercises had to be made very visible in order to avoid this history hampering the SPI project. Fortunately many of the glaring problems with the implementation of the Infomet methodology, such as inflexibility in its application on diverse projects, were well understood, and great pains were taken to avoid these mistakes during the SPI project. (In this case, every department and project was encouraged and assisted to customise each software engineering process or practice according to its own circumstances and needs.) In this way, the lessons learnt during the Infomet methodology implementation contributed to the work of the SPI project.

The diversity of technical knowledge and activities in general differed vastly between the SDD and the business departments. The SDD builds and maintains a number of different types of applications (e.g. mainframe, client/server, PC and data warehouse-based applications), applies many different information and communication technologies and has been exposed to a number of new software development tools and technologies in recent years. For example, over the past few years the SDD has been actively involved in the implementation of new data warehousing and data mining tools, has implemented new development languages, and has utilised technologies such as call centres, Integrated Voice Recognition (IVR) and the Internet. This has effectively meant the SDD's staff have been widely exposed to a wide range of technologies. The business departments, on the other hand, tended to have had a more narrow exposure to technology. Their staff also seldom had an IS background and did not generally have that wide an exposure to technology in general.

The SEPG also had a wide exposure to both related subject matter and technology in general. Not only was the educational level of the SEPG generally quite high (they all had at least one, and in many cases multiple tertiary qualifications), but they had also had a large exposure to outside experiences. In fact, 70% of the SEPG staff were appointed from a diverse range of different IS organisations, and industries in recent years. The group, by virtue of their role in the IS division, also interacted widely with other IS departments, external organisations and consultants on a regular basis. In this way, the SEPG was able to introduce a fairly diverse set of experiences to the SPI initiative, thereby facilitating the learning process.

In summary, the learning-related scale, the related knowledge and, at least in the case of the SDD and the SEPG, the diversity of technical knowledge and activities in the areas covered by the SPI project, were such that an initiative of this nature was more likely to be initiated and sustained.

e. Management Orientation

Top management's orientation toward time, market and customers can result in distinctly different approaches to TQM, and these approaches, in turn, influence TQM's chances of success (Choi and Behling, 1997). Since, as was pointed out in the literature study, SPI is a form of TQM applied to the software organisation, it follows that top management's orientation will also affect the approach to and the likelihood of success in an SPI initiative.

Contradictory observations were made of the management orientation in the departments covered by the case study. On the one hand, the head of the SDD seemed to have a predominately developmental orientation which, according to Choi and Behling's (1997) research, would indicate a higher likelihood of an SPI project being initiated, actively pursued and sustained. A developmental orientation is one in which management is a) mainly concerned with growing their business and client base over time by being a "world-class" software organisation, b) focused on improving current performance so that the organisation can be better positioned to compete in the future, and c) committed to seeing the customer as a partner in a co-operative relationship.

Indicators that the head of the SDD had a developmental orientation include:

- The fact that some of his stated objectives for embarking on the SPI project in the first place were to develop the SDD *"into a world class IS organisation that is the preferred IS vendor to the marketing division and whose staff perceive themselves as a professional team"*
- His view that the business departments participating in the SPI project were his partners in the endeavour to meet the marketing division's technology requirements in the future. He underscored this view by making every effort to involve the heads of his client departments in the SPI initiative and to work with them in order to improve their combined ability to satisfy their ultimate clients. This is also evidenced by the fact that these managers were all active co-sponsors of the SPI project.

However, almost all the other observations indicated that most of the other managers impacted by the SPI project did not entirely share the SDD head's developmental orientation. On the whole, managers appeared to feel significant pressure in keeping up with their customer demands and felt that their clients were not, in fact, generally not satisfied with their performance. Related to this was some tension between the SDD and the business departments involved in the SPI project. A common complaint was that the SDD did not deliver software solutions fast enough and that the business departments, in turn, did not do *"their bit in terms of providing the requirements and made impossible demands"* on the SDD.

The net result of this was a general sense that, despite a mutual dependency on each other, an "us versus them" sort of relationship existed between the SDD and the two business departments. There was also a sense that the departments, particularly the SDD, were fighting for survival with the very real threat of being outsourced or bypassed in favour of purchased packages.

The departments in the SPI initiative also found themselves in a situation where they spent much of their time and energy "fighting fires", dealing with operational issues, and desperately trying to keep up with their clients' demands. There was very little evidence of a long-term, strategic emphasis to develop a vision for the marketing division's technology needs.

The above findings would indicate that the three departments involved in the SPI project had either a Tactical or a Defensive Management Orientation according to Choi and Behling's (1997) model, and would therefore be less likely to sustain a comprehensive SPI project over an extended period of time.

This contradiction between the apparently developmental orientation of the SDD head and the tactical or defensive orientation that seemed to dominate most of the other role-players, could explain why the main energy and drive to initiate the SPI project came from the SDD head and why, certainly in the early stages of the project, the business department heads were more reserved in their support for the SPI project. However, once the SPI project was running, the tactical or defensive orientation of the rest of the role-players could explain why there were constant battles to make time available for the SPI project in the face of operational project pressures, why the SPI project, in the words of the project manager was "*always seen as a secondary priority and an additional load over and above work that must be done*", and why the project was not sustained when faced with the uncertainty of the company's restructuring.

iv. Change Context

The context within which an SPI project occurs can provide the opportunities and stimulus for change or can act as a barrier to effective change. The observations of the case study identified contextual factors in both of these categories.

The company within which the case study was performed was operating in a very competitive environment. Threats and challenges facing the South African financial services sector at the time included an increased threat of global competition, increased domestic competition from new companies and products, and changes in consumer preferences for the types of financial products that they bought. As a result pressure existed in the company to increase revenues, cut costs, work more efficiently, and deliver increased customer value and the departments participating in the SPI project found themselves in a context where they had to improve their performance.

What is more, the use of information technology is regarded by the company as being critical to its success. This was highlighted by the Executive Chairman of the company in a recent meeting with all IS staff where he stated that *"IT is extremely important to the future success of this company"*. This sentiment was echoed by the head of one of the business departments in the SPI project who felt that the marketing division of the company *"would be left behind"* if it do not use IT appropriately. The pressure to improve the performance of the company and the high dependence on IT is indicative of a context that is conducive to an SPI initiative.

At around the same time that the SPI project was beginning a company-wide programme aimed at improving organisational efficiency and effectiveness was launched by the Managing Director. This programme, the Workplace Improvement Programme (WIP), had strong TQM roots. An analysis of the WIP by the SEPG found that between 60% and 70% of its requirements could be satisfied by the implementation of SPICE in a software development context. The departments participating in the SPI project therefore saw the project as helping to satisfying the requirements of the WIP. At the same time, their investment in the SPI project was seen to be further justified by the existence of this complementary company-wide programme.

In terms of organisational structures, the SDD and business departments participating in the SPI project were in separate organisational units and had their own management teams, procedures and practices. At times this resulted in differences of opinion between the different departments as to which software processes and practices should be implemented first, and who should be responsible for which parts of the different processes. The ownership of the project management process was one such area of contention.

Separate organisational structures also meant that, on occasion, the SPI project found it hard to get the departments to agree on a consistent way of applying a given process or practice or to co-ordinate their efforts to realise the changes required by the SPI project. For example, in the implementation of the new approach to requirements analysis, the business departments were not prepared to provide as many models as the SDD would have liked.

The structure of the three departments involved in the SPI project resembled centres of excellence. The departments shared a single project office which consisted of a group of project managers who were only responsible for project execution. The projects were staffed from the teams in the departments which were generally arranged around a particular technology and client grouping. For example, the one business department mostly consisted of analysts that focused on a single class of business application, while the software development department's teams were generally organised around a single type of technology.

This centre of excellence-type organisational structure did turn out to be helpful to the SPI project and made the implementation of new processes and practices easier in some cases. For example, the introduction of the RAD methodology was chiefly targeted at the project management group (because the focus at that stage was on defining projects incorporating a RAD life cycle and project structure); the object-oriented analysis techniques were specifically targeted at the PC development team and one of the business departments; and the relationally-based analysis techniques were targeted at the management information team and the other business department.

An aspect of the IS division's organisational structure that was seen to be very beneficial to the SPI project was the presence of a strong corporate SEPG. The SEPG, with its collection of well-defined software engineering processes, was regarded by all the respondents as being critical to the success of the SPI project, especially as the ability of the departments' staff to implement these practices without outside help was low.

There were very few formal IS policies and associated management behaviours that made the pursuit of SPI initiatives an imperative in the IS division. An exception was of the establishment of the corporate SEPG and the clear support for this sort of work by the CIO. This meant that, with the exception of personal motivation, there was very little pushing senior management in the IS division to initiate and sustain an SPI project.

Therefore, although the SDD head enthusiastically drove the establishment of the SPI project, it was hard to sustain the project in the face of operational project pressures, changing company priorities and the restructuring of the company that resulted in the removal of the SDD head's authority to promote and "enforce" the continuation of the SPI project. In other words, key management role-players could no longer sustain the project in the face of the restructuring since they were placed in a position where they were less able to fulfil their sponsorship role.

The company's restructuring, and the impact that it had on the main role-players motivating the SPI project was given as one of the main reasons that the SPI project was terminated early. The presence of appropriate policies regarding the on-going application of SPI in the IS division may have helped sustain the SPI project, despite the removal of its immediate sponsors.

In summary, the change context was found to support the SPI project in the following ways:

- Strong competitive pressures and client demands for the rapid delivery of high quality software contributed to the motivation behind the establishment of the SPI project.
- The presence of a company-wide improvement initiative supported and complemented the SPI project.
- The existence of a project office and the "centre of excellence" nature of the organisational structure of the three departments involved in the SPI project made the work of the project a bit easier.
- The presence of a strong SEPG was an important enabler of the SPI project.
- Although not discussed in this section, the ability of the organisation to overcome learning barriers described earlier, also made it feasible to embark on the SPI project.

On the other hand, the SPI project was hampered by the following elements in the change context:

- The fragmented organisational structures between the three departments involved in the project created gaps in the software processes and resulted in communication, prioritisation and co-ordination problems.
- A lack of formal IS policies and associated management behaviours, whose presence might encourage SPI projects in the company, made it hard to motivate and sustain the project.
- The massive organisational change in priorities and structures removed the direct influence of key role-players required to sustain the SPI project in the absence of other sustaining forces.

4.2.2.2 Change Implementation

In terms of the proposed SPI framework, the implementation of change is discussed in three categories: software processes, non-process change dimensions and change management.

i. Software Processes

The literature study described a number of areas regarding software processes that were of interest to the case study. Included were the manner in which the SPI project approaches SPI, the software engineering processes and practices targeted for implementation, the software process maturity of the departments in question and the extent to which the SW-CMM or SPICE models were applied. These areas are explored in this section.

The SPI project was strongly influenced by both the SW-CMM and SPICE models, since the SEPG promoted the use of both models in a complementary way. This mix-and-match approach to using the two SPI models resulted in the SEPG adopting what they liked from each model and downplaying other aspects. For example, SPICE, and not SW-CMM, assessments are emphasised, whereas the SW-CMM, and not SPICE, is used to explain the concepts of process maturity to IS departments. This meant that neither of the models was applied in its entirety by the SPI project, but rather that they were both used to give pointers and guidance to the project.

One of the reasons that the usage of both SPI models in this way did not seem to present any problems to the SPI project, or cause any untoward misunderstandings by the staff impacted by the SPI project, was probably that the project focused mainly on "implementing software engineering" and not necessarily on achieving a given software process maturity level. In other words, the emphasis of the project was to identify and implement those software engineering processes and practices that were deemed to add value to the departments' day-to-day work. The SPICE and SW-CMM models were generally only used to help identify those processes and practices and provide inputs to their implementation.

Despite the lack of emphasis on obtaining a higher software process maturity level, the low maturity of the departments at the start of the SPI project was used to help motivate and justify the project. At project's inception the departments in question had an overall software process maturity corresponding to level 1 of the SW-CMM (the Initial Level). Individual software process capabilities all corresponded to level 0 (Incomplete) or level 1 (Performed) in SPICE model terms.

The software process maturity level was determined from (1) a SPICE assessment of a typical project, (2) an analysis of the departments' processes using BPR-like techniques, (3) a process assessment using the Process Advisor product, and (4) anecdotal evidence of SW-CMM level 2 processes not being consistently applied or being subject to documented departmental policies and procedures. The Process Advisor product provides a questionnaire-based assessment method that is based on the SW-CMM. It contains about 100 questions and covers the SW-CMM up to level 3. This assessment was applied to all the staff in the three departments at the beginning of the SPI project and strongly confirmed that the departments fell somewhere between levels 1 and 2 on the SW-CMM maturity scale.

Based on the analyses and assessments described above, a study of the SPICE and SW-CMM models, and a number of workshops with the staff of the three departments participating in the project, a number of software engineering processes and practices were identified for implementation by the SPI project. These included:

- Requirements analysis practices.
- Project Change Control, together with some very elementary configuration management practices.
- A project health check questionnaire used to determine whether the basic software engineering and project management practices were applied, to provide project managers with a checklist of critical software project practices and to determine training needs. (This largely tested SPICE level 1 & 2 and SW-CMM level 2-type practices).
- Project planning and tracking
- Project estimation for new development using function point analysis.
- Project and task prioritisation.
- A form of document management to help manage a project library.
- Quality planning for projects.
- The implementation of a RAD methodology
- Reuse (This was terminated early. See the discussion in section 4.2.2.2.ii.e. below.)
- Reviews
- The use of the Process Engineer software process management tool
- The use of a new database containing the IS divisions standards.

Although level 2 software process maturity was not explicitly targeted by the SPI project, many of the practices included in this list corresponded to those required to achieve a level 2 maturity rating according to the SW-CMM (five out six Key Process Areas were addressed in one way or another).

In conjunction with, and in support of, the implementation of the targeted software engineering processes and practices, the SEPG was commissioned to consult on software projects in the departments in order to ensure that their expertise was on hand as and when it was needed. This helped ensure that the project managers and teams had immediate access to support when they found themselves struggling to apply the new processes and practices.

In implementing the above software engineering processes and practices the SPI project adopted an approach that, in one way or another, looked very similar to the eight SPICE SPI steps described in figure 2-4. (1) *The need to embark on an SPI project was identified* by the software development department head; (2) *an SPI project was initiated* in response to this need in conjunction with the two business departments and the SEPG; (3) *analyses and assessments* of the involved departments' software processes were performed; (4) *the results were analysed* to identify required software engineering processes and to draw up the detailed SPI project plan and organisation; (5) the required improvements in terms of new processes and practices were *implemented* and then (6) monitored to *confirm* their effectiveness and overcome any initial problems; (7) structures, tools and procedures were put in place and responsibilities defined to *sustain* the new processes and practices; and (8) additional SPICE assessments were planned to *monitor* the overall improvement status of the departments. Only this last step was not performed as the SPI project was terminated before the planned SPICE assessments could be performed.

Within this overall approach, the SPI project defined four implementation cycles to help it focus on sub-sets of processes and practices at a time and to provide interim milestones by which to track and evaluate progress. In this way each implementation cycle addressed the fifth and sixth SPICE SPI steps for a few of the required software processes and practices.

In conclusion, although not explicitly following the SW-CMM or SPICE models, the SPI project followed many of their guidelines. What is more, they tended to emphasise typical level 2 processes and practices which is consistent with a level 1 organisation trying to move to a level 2 software process maturity.

ii. Non-process Change Dimensions

The expected interactions between the software process and other change dimensions at levels 1 and 2, as described in table 2-10, are the focus of this section. Processes and practices that could have led to higher software process maturity levels were generally not addressed in the scope of the SPI project and it is therefore difficult to comment on the validity of the interactions described at higher levels in table 2-10. However, where meaningful observations as to the validity of the model in table 2-10 were made, they are described.

a. Structures

The literature study identified the following aspects of organisational structures that may impact an SPI project: the impact of organisational designs, such as centres of excellence, a focus on team-based practices, and organisational structures directly involved in the SPI project itself.

The centre of excellence-type organisational structures of the three departments, previously discussed in section 4.2.2.1.iv, were found to facilitate the implementation of some of the required software engineering processes and practices. In particular the presence of a project office, with its complement of project managers, was found to be particularly useful in implementing many of the practices required to move to a level 2 software process maturity.

Also, although many of the elements required to make centres of excellence function totally effectively are only typically present at a level 3 maturity, the presence of centre of excellence type structures, especially the grouping of similar skill requirements in teams, seemed to make the implementation of certain processes and practices easier. This may follow from the fact that grouping people with similar skills and needs may facilitate the creation of a focused learning environment for that group that is not distracted by the broader range of needs found in a more diverse grouping of staff. Also, these focused teams helped the SPI project to target training, development and improvement activities more precisely, speeding up the change process and reducing complaints that the project was wasting staff time.

One can therefore identify a number of different teams within the organisational structures of the departments studied. These were made up of (1) the project managers in the project office, (2) the people within the departments, grouped according to tasks and technologies, and (3) temporary project teams created to undertake new software development and large scale maintenance.

These teams seemed to have a lot of freedom to determine the way in which they performed their work. As discussed in the section on cultural readiness, it was found that management tended to trust its staff to make technical decisions and changes to their own processes and practices without undue interference. However, responsibility for organisational and management issues such as planning, progress reporting and resource management remained strongly vested in the hands of team leaders and project managers. In this sense the teams were therefore not entirely self-directed.

Nonetheless, the technical autonomy of the teams to change the ways in which they worked was found to contribute positively to the SPI project. The teams played a major role in defining the detail of the processes and practices that they wanted to use, fine-tuned the processes and practices to suite their own specific contexts, and had a sense of ownership over their work practices. This ownership was displayed in a situation where an analyst insisted on being allowed to try to apply the new analysis practices despite management's concern that the use of the new methodology was slowing the project down. The spontaneous definition of some elementary code reuse practices by one of the project teams is another example form the case study of how members within teams could contribute to software process improvement and innovation. This team's initiative lead the SPI project to recognise the opportunity to broaden the application of reuse throughout the SDD. The fact that this failed (see the discussion of reuse below) did not detract from fact that the teams had the potential to contribute positively to the SPI project.

The various role-players in the SPI project have been described previously in the change resources section. These were the SEPG, the SEIG, process jockeys from the SEIG, a management steering committee, and organisational development expert. Each of these roles was found to play an important part in the SPI project, with the management steering committee providing the management sponsorship and addressing prioritisation and commitment issues, the SEIG and the jockeys implementing the new practices, the organisational development consultant keeping change management and "soft issues" on the agenda, and the SEPG providing critically required expertise and direction on the project.

The SEPG was regarded by most respondents as being very important to the success of the SPI project as the SEIG relied on the software process expertise and guidance that they provided. This would appear to indicate that the SEPG, usually an element of level 3 in the SW-CMM and SPICE models, was beneficial at level 1. Although the emphasis of the SEPG would broaden at level 3 to include an organisation-wide perspective, and even though the SPI project was largely focused on level 2 type activities, the departments involved in the case study found that the expertise and focus of the SEPG was critical to the project's ability to perform.

The findings of the case study therefore support the expectations raised in the literature study that self-directed work teams and SPI project roles, such as the management steering committee, process owners and a process improvement project team, are likely to assist an organisation engaged in level 1 and 2 SPI activities. In addition, the study also found that centres of excellence (in the form of groupings of skills and the use of a project office) and the presence of a strong SEPG also contribute strongly to an organisation attempting to mature its software processes from level 1 to level 2.

b. People

An organisation's "people" practices in terms of recruitment, training, development, and rewarding its staff can have an impact on the success of an SPI initiative.

The roles and responsibilities of staff in the departments participating in the SPI project had not been formally defined at the time of the project. This implies that staff were neither recruited nor explicitly trained with the departments' software processes and practices in mind. Most recruitment of staff in the SDD was based only on the technologies used, for example, Visual Basic and object-oriented design, or the potential of people to learn technical skills (as determined by aptitude tests in the past). Staff in the business departments usually had some experience in the business itself before beginning work in the technology-related parts of the marketing division.

The lack of training plans meant that training was generally provided on an ad hoc basis. Also, most of the training provided focused on developing an individual's knowledge in the particular software technologies or, in the case of the business departments, business areas, that people needed to perform their day-to-day tasks. This finding is consistent with many of the aspects of Baldwin et al's (1997) Stage 1 learning strategy which may be expected in organisations with ad hoc processes.

Training directly related to SPI itself was generally limited to the software engineering processes and practices that were being implemented by the SPI project. Therefore, other than those few aspects addressed during the initial presentations at the start of the project, very little general SPI and quality management training was presented. This was probably one of the reasons that there was a general lack of understanding of the overall discipline of SPI and the long term nature of an SPI project. It would also help to explain why the departments' operational focus tended to result in the neglect of the SPI project in the face of project pressures, since the long-term, "big picture" vision for SPI was not properly communicated to the staff.

The above recruitment and development practices would seem to be consistent with the conclusions of the literature study regarding an organisation with a lower software process maturity. However, during the course of the SPI project three developments took place in this area to indicate a move towards practices consistent with a more mature organisation.

- Firstly, the software development department recognised the need to define and develop specific roles. The intention was that staff would be assigned to one or more roles and that training would be provided in order to help the individuals fulfil these specific roles. This was highlighted by the head of one of the business departments who regarded formalising the department's software engineering processes and practices and the associated roles as being *"very useful to train and develop people in terms of how we work"*.
- The second development was the insistence of the steering committee that all training provided in the software processes and practices be accessible to them so that they could easily obtain training for new staff members in the future. The SEPG therefore ensured that the training was readily available, either through the company's training department or from external training vendors.
- The third development was a growing clarity about the roles in some of the software engineering processes and practices implemented by the SPI project. Of particular note were the project management and analysis processes where there was an increased understanding of what was expected from the staff fulfilling the required roles and their training requirements. These are level 2 processes and it is arguable that their implementation was enhanced by the definition of the project manager and analyst roles and responsibilities, and the recruitment and training of people in terms of these definitions. In this way, skill-based recruitment and training in terms of the organisation's software processes can already begin at level 2, and not only at level 3 as indicated by the literature.

Finally, there was very little evidence of performance appraisals and rewards being tied to the attainment of SPI objectives. The reasons given for this were that the company rewarded outcomes (for example, the delivery of a system more quickly than expected) rather than meeting SPI objectives (inputs or the means to these outcomes).

The SEPG argued against this as they felt that it detracted from the SPI project and the institutionalisation of the required software processes and improvement practices. They argued that the rewards policy would mean the SPI project would always be neglected in the face of operational project pressures. Pure outcomes-related rewards, such as rewarding the rapid delivery of systems, fail to motivate an SPI initiative because the improved outcomes often lag behind the improvements introduced by an SPI initiative by years. Short-term measures, such as simple bottom-line measures of financial performance, will tend to undermine initiatives investing in the future performance of a company (Kaplan & Norton, 1992).

One exception to the lack of rewards tied to the SPI objectives, which began to emerge towards the end of the SPI project, was in the project office. Here the project office manager decided to make the practice of project management related processes, such as project change control, an input into all project managers' performance appraisals. However, this was still the exception rather than the norm and was only being measured on a subjective basis by the project office manager.

In summary, the case study concurs with the conclusions of the literature study in a number of ways. Firstly, it largely confirms that competency-based staff recruitment and training is only likely to be utilised at higher software process maturity levels than that of the departments covered by the SPI project. Secondly, the overall learning strategy of the departments was as expected for an organisation with a low level of software process maturity. Thirdly, as was expected, the lack of general SPI training and SPI-related rewards undermined the SPI project.

On the other hand, the literature had expected that the definition of roles and the recruitment and development of staff in terms of the implied competencies would only be prevalent at level 3. However, the case study highlighted the fact that it is possible to begin applying this practice at level 2 for those processes that directly contribute to the achievement of this level of process maturity. Examples of these roles, in SW-CMM terms, are the project manager and analyst roles.

c. Information and Technology

The use of two classes of tools in the context of SPI were investigated. The includes those tools that provide automated support of software engineering processes and includes programming environments, CASE tools and testing tools. The second is those that directly support the practice of SPI.

The use of automated tools for development and maintenance was generally focused on tools directly related to the development of program code. For example, the Microsoft Visual Basic environment, editors, compilers and a debugging tool for mainframe COBOL programming, and a tool for developing data warehousing-type applications were predominately used. CASE tools were used on a very limited and ad hoc basis for modelling. A configuration management tool was utilised to manage source code on more recent Visual Basic projects. Microsoft Project was fairly widely used by project managers to aid project planning and tracking.

The use of CASE tools to support modelling during analysis and design had generally been a disappointment with the perception that CASE, although useful as a diagramming aid in some cases, failed to add any real value. The CASE tools purchased by the departments had generally fallen into disuse fairly soon after the fanfare associated with their purchase had passed. Respondents ascribed this to a number of factors, including the fact that the tools were implemented to support a methodology that was not fully adopted by the analysts, the perception that CASE was not relevant to maintenance work, and that the developers tended to emphasise writing code and did not place much value in the models delivered by the analysts.

The case study is therefore fairly consistent with the findings in the literature that CASE is less likely to be successfully adopted in the absence of an established methodology. This is the situation in the departments studied which did not have a well-defined and consistently applied methodology in place for analysis and design.

On the other hand, an increase in the use of project management and the software configuration management tools was observed during the period covered by the SPI project. This could be partly attributed to the increased focus on software processes that made their usage more attractive. For example, the use of the project management tool was partly motivated by more formal project planning activities. The use of the configuration management tool was partly driven by the fact that it facilitated the practices of a project team working closely together on a single system and their attempts to manage code-level reuse with the project.

The finding that tools supportive of the processes required to move to level 2 software process maturity were more actively used is consistent with the expectation represented in table 2-10. It indicated that tools supporting level 2-type processes, including project management and software configuration management should be utilised at this level.

In terms of tools to support SPI, the only tool used in the departments was the Process Engineer software process management toolset. This was used to publish and distribute the methodology and other software processes to all members of the departments. It was also used by project managers to help develop project plans based on the defined methodologies. Despite the fact that the Process Engineer initially overwhelmed some project staff with its apparent complexity, the toolset was eventually positively received by project managers and those people that made extensive use of the methodologies published in its process repository. The major benefits of Process Engineer were that it made the methodologies freely available to all staff in a very accessible, on-line, format and that it helped project managers produce large, comprehensive project plans, based on a sound methodology quite quickly.

On the other hand, the SPI project did not use Process Engineer's capability to gather process data from executing projects. Process Engineer was therefore not fully exploited as a software process management tool. This is not entirely unexpected since gathering process data of this kind is typically only expected of organisations pursuing a level 4 software process maturity.

In conclusion, Process Engineer was regarded as contributing positively to the SPI project as it helped establish the methodology and software processes in the departments. This software process management tool was therefore found to be valuable at a lower level than the literature would necessarily have expected and should possibly have been aligned with level 1 in figure 2-10. At level 1 the tool is most useful as a way of sharing software process knowledge and experiences by making them available on-line. At level 2 the tool could be used by project managers in developing project plans. At level 3 it would be useful for documenting and disseminating the organisation's standardised software processes.

The other findings described in this section, namely that CASE tools struggled in the absence of a well-defined methodology for analysis and design typical of a level 3 organisation, and that tools supporting level 2 type processes should be aligned with level 2 SPI efforts, are consistent with the conclusions of the literature study.

d. Management

Management practices can facilitate or hamper the practice of SPI in many ways. Within the context of the SPI project a number of management practices were seen to contribute positively to the SPI project. Areas such as management's commitment, their insistence on the use of certain practices, and the use of a steering committee, have been discussed.

Areas that could have been used by management to support the SPI project, but were not, included the use of rewards to encourage the attainment of SPI objectives and the use of measures and benchmarks. The lack of rewards has been discussed previously.

There were no indications that measures and benchmarks were effectively used by management to support SPI. The only formal measures used by the departments engaged in the SPI project were internal Value For Money measures used to track their budgetary performance. No other measures were used to track and improve software process performance. The major reasons identified for this lack of measurement were that the collection of the required measures was perceived as being too large an overhead and that the use of measurement is not a typical part of the company's management practices. This lack of use of even performance loop information is not what one would expect of an organisation formally engaged on an SPI project, even if it is still attempting to obtain a level 2 maturity, and made it impossible to track (in a formal manner) the improvements in performance brought about by the SPI project.

Benchmarks were also not formally used by the SPI project to compare the departments' software performance and maturity with any other organisations. Besides being hampered by the same problems as the use of measures, benchmarks of software process maturity could not be effectively used as there had not been sufficient self assessments by the time that the SPI project was terminated. Useful benchmarks of the performance and maturity of similar organisations were also hard to come by.

However, the lack of benchmarking by the SPI project is expected, given the low software process maturity of the organisation, but the SPI project under study should at least have made use of performance loop information. This would have made it easier for management to quantify the improvements brought about by the SPI project and would also have helped practitioners to control the execution of processes better.

The assessment of management's use of appropriate strategies, structures, systems and practices to facilitate the SPI project is mixed. On the one hand areas such as providing sponsorship and resources for the project can be seen to be positive. However, management failed to make use of SPI-related rewards. They also did not ensure that staff had sufficient time available for the SPI project and made no use of proper performance measurements. Finally, management was perceived, at times, to be passive "cheerleaders", rather than active facilitators of the SPI project.

e. Reuse

Reuse is seen by many authors as a way of improving software development performance (for example, see Henry & Faller, 1997; Lim, 1994; Short, 1997). Therefore, during the SPI project an attempt was made to institutionalise software reuse. This happened after one of the development project teams implemented some reuse practices within their project in order to reuse common code components and to capitalise on components that had been built on previous projects. The team's efforts entailed the establishment of some basic reuse standards and a simple set of reuse library procedures. Encouraged by that project team's reuse successes, the SPI project decided to capitalise on their work and implement reuse practices across the entire SDD.

The major drivers behind this were (1) the need for greater development productivity, (2) the desire to increase reuse between applications (particularly since many of the department's applications form part of greater integrated application suites), and (3) frequently rewrites or updates to the applications in response to changes in the company's products or legislation. Most of these changes involved, in effect, a variation on a previous version of an application and the developers therefore perceived many opportunities to reuse significant portions of these earlier versions.

The SPI project, however, abandoned its efforts to formalise and institutionalise the project team's reuse practices throughout the SDD as it was quickly discovered that many of the enablers required for systematic reuse (i.e. reuse applied across a large group of projects over an extended period of time in a concerted way) had not yet been completely implemented in the department. In particular, respondents mentioned the absence of configuration management, well articulated requirements, sound software quality assurance and the consistent use of a reuse-enabled methodology. This confirms the literature study's conclusion that it is unlikely that systematic software reuse would succeed at an organisation-wide level if many of the level 2 and 3 maturity practices are absent, as was the case in the departments under study.

On the other hand, the practice of ad hoc or project-based reuse is feasible in the context of a single project team at all software process maturity levels. This was shown by the project team described above. However, this brand of reuse is not likely to be scaleable across a number of projects in a sustainable way without the addition of many of the enabling software processes that will provide additional control and visibility over the reusable objects.

One discipline not explicitly addressed in the literature, the absence of which was considered by the respondents to have a negative affect the implementation of reuse, was the lack of a requirements analysis discipline. The project team concluded that this was a necessary enabler of reuse as it facilitated the identification of reusable components and would have enabled the reuse process to begin much earlier in the software development life cycle. This conclusion does not contradict the expectation that systematic software reuse is best attempted in a level 3 organisation since requirements management is a level 2 type practice.

f. Enterprise Architectures

Although the departments in question understood the need to utilise enterprise architectures for planning and overall system design and integration, no evidence was found that they formally made use of enterprise architectures during the time of the case study. This does not contradict the conclusions of the literature study which had pitched the use of enterprise architectures as an enabler or facilitator of SPI at level 3.

g. External Alliances

An organisation could increase its own software process maturity level by working with partners or vendors that have a higher maturity than itself. However, the departments covered by the case study had only recently begun to interact more closely with external IS vendors. Prior to that most vendor contacts were at an arms length and generally focused on the purchase and technical support of tools. These vendors therefore only had a minimal impact on the software development practices of the SDD and did not assist the SPI project to increase the IT capabilities of the department.

Although it was too soon to comment on the long term impact at the time of the case study, two recent engagements with vendors appeared to offer some possibility of affecting the departments' IT capability. One vendor provided a new data warehousing tool, and at the same time provided value-added services in the form of a methodology and associated training that directly supported the tool and its effective usage. This could certainly contribute to increasing the capabilities of the teams utilising the tool.

The second engagement involves the selection, purchase and joint customisation of a large software application. The organisation that is providing the package clearly has fairly well-defined and mature practices in the areas of application architectures, development practices, project management and requirements management. The exposure that the software development and business departments will get to these practices could, once again, prove to be a valuable contributor to increasing the organisation's software process maturity. In fact, this was seen as one of the positive aspects of buying the application at the time.

In summary, although too soon to comment meaningfully, the findings of the case study would seem to confirm the possibility that close working relationships with organisations that possess a higher software process maturity, or have knowledge of more mature software engineering processes and practices, can help "pull" a less mature organisation towards more mature practices.

This concludes the discussion of the non-process change dimensions evident in the case study and how they are related to the software process dimension in the ESPIM.

iii. Change Management

An SPI initiative typically requires a significant change in the software organisation's culture and the ways in which people view their work outcomes and people perform their tasks. Change management strategies and actions are often used to anticipate and overcome resistance to this change, thereby facilitating the realisation of sustained changes. This section explores the change management strategies of the SPI project in the departments covered by the case study.

Change management issues were explicitly addressed by the SPI project in a number of ways. One of the company's internal organisational development experts was assigned to the project team with the explicit purpose of implementing a change management strategy and keeping "soft issues" on the project agenda.

Elements of the change management strategy included:

- Performing a *stakeholders analysis* and developing a *communication plan* targeted at the different stakeholder groupings.
- Ensuring proper management *sponsorship* through the inclusion of senior managers from all affected areas on the steering committee, and educating team leaders and the jockeys from the SEIG on their responsibilities as second-level sponsors responsible for sustaining the changes in their immediate areas of influence.
- *Keeping all staff informed and involved* in the change through initial presentations outlining the expected benefits of the SPI project, soliciting their inputs on prioritising the software processes that were to be addressed by the project, and providing feedback at the end of each project implementation cycle on the cycle's outcomes, successes and problems.

- *Soliciting real feedback* from staff through encouraging open debate on the project in the SEIG, frequent informal discussions with staff on the ground and the use of a formal survey administered by the organisational development expert to identify issues and concerns and foster open discussion. An example of the informal discussions used to solicit feedback was the practice whereby members of the SEPG would visit members of the departments affected by a new software engineering process or practice to find out how well it was working, what their problems and suggestions were, etc.
- *Involving staff in the actual definition and implementation* of the new software engineering processes and practices. To facilitate this the SEPG tried to avoid presenting a final solution, but rather proposed an outline or template for a new process or practice and then encouraged and assisted the departments to customise and implement a version of the processes that best suited their particular needs. It was intended that this, together with the use of jockeys from the departments themselves to drive the implementation, would result in greater buy-in to the changes.

Most of these change management elements were directly incorporated into the SPI project plan and procedures, and were therefore managed as an inherent part of the project.

Another aspect of the SPI project plan and procedures was the recognition that it was impossible to plan all the detail of the project in advance, given the fact that this was the first full project of its kind in the company and the environment was rapidly changing. The project plan therefore only formalised the structure of the project in terms of project implementation cycles, roles and responsibilities, and a first cut proposal for which software engineering processes or practices should be included in each cycle. The detail of a cycle was only planned at the time that it was due to start. In this way the latest experiences, usually based on a review of the previous project implementation cycle, and current circumstances could be taken into consideration by the project as it progressed.

This flexibility was further enabled by the fact that the SEIG and the steering committee allowed frequent changes to the SPI project's content and focus as the project progressed. For example, when it was realised that the original content of a project implementation cycle covered too many practices for the staff to absorb given their workloads, a decision was made to drop some of the practices from that particular cycle and focus on a few critical processes. They also allowed the reuse efforts to be terminated quite quickly once it became apparent that it was not appropriate to be focusing time and energy on reuse until some of the enabling software engineering processes and practices had been implemented.

The flexibility built into the project also made it possible for the SPI project team to respond to opportunities to introduce unexpected changes that could add real value. For example, while assisting a project manager to apply the new RAD methodology, it became clear that the project manager was struggling to keep track of, and manage, all the project documentation that was being produced. This presented the opportunity to introduce the project library document management procedures. This required relatively little effort to implement as the SEPG had access to a defined set of procedures for project library document management. However, their introduction added real value to the project and was seen to be an important contribution from the SPI project to the functioning of the project office. This example is an instance of an *opportunity-based* change, that is, a change that is not anticipated ahead of time, but is introduced purposely and intentionally during the change process in response to an unexpected opportunity or event (Orlikowski & Hoffman, 1997).

In terms of the other types of change defined by Orlikowski and Hoffman (1997), the SPI project also recognised *emergent* and *anticipated* types of change. The previously described work by a development project team in the area of reuse is an example of an emergent change, that is, a change that arose spontaneously from local innovation and was not originally intended or anticipated. The implementation of all the pre-defined software engineering processes and practices is an example of anticipated changes.

In both the emergent and opportunity-based change examples given above, the SEPG's capability to respond quickly to unanticipated changes was a potential source of strength to the SPI project. A flexible and agile SEPG with access to a range of useful resources can therefore be seen as a key enabler of this sort of change on an SPI project.

The SPI project therefore took change management considerations into account and was structured to accommodate both anticipated and unanticipated changes. This was further enabled by the flexibility allowed by the steering committee and the SEIG, and the SEPG's ability to respond quite quickly when a change required the introduction of different processes or practices.

However, the organisational development expert involved in the project did raise the concern that, although the SPI project did all the correct things in terms of *formal* change management activities, the energy and action to re-inforce the changes in *informal* ways seemed to be lacking. Examples were that senior management seldom reinforced their steering committee role through spending unstructured time discussing the progress and issues of the SPI project with the "people on the ground", and that they seldom used rewards, even informal rewards such as verbal affirmation, to encourage the realisation of the changes.

4.2.2.3 Change Outcomes

The last area explored by the case study addresses the outcomes of the SPI project. At the outset of the project the expected outcomes were predominately focused on the need to increase performance in terms of software delivery. In other words, of the three types of outcome that could be expected from an SPI initiative (increased software process maturity, improved process outputs and improved quality of work life of employees), improved process outputs seem to have been the greatest driver behind the project. In particular, the expectation was that the project would increase productivity and decrease time to market for software products, while improving the quality of the products. This was accompanied by a sense that the respondents wanted a "*better way of doing things*" - an objective that was never formally articulated during the project.

Since no real attempt was made to define any of the expected outcomes or specify how they would be measured, it was impossible to determine with certainty whether these outcomes were realised or not. Also, the fact that the project had only run for a relatively short period of time in SPI terms, meant that it could not reasonably have been expected to produce spectacular results in increased software process maturity and other process outcomes. This is evident from the fact that none of the respondents could point to improved software project productivity or quality at the time of the case study. They did, however, identify a number of other positive outcomes. The following points were raised:

- The business departments involved in the SPI project expressed an increased level of trust in the abilities of the SDD to meet their needs.
- The business departments felt that they could better articulate what they expected from the SDD.

- Related to the above is the fact that the business departments felt that improvements in project planning and tracking, project change control, reviews and analysis all contributed to improved insight into project progress and control over the projects' results.
- A number of staff members felt that they had obtained answers to issues that had been concerning them about how to go about their day-to-day jobs and felt *"ecstatic at having a right way of doing things"*. One of the business department heads expressed this well when he said that he could *"see that people are more focused and they know more of what is expected from them"*.
- Closely related to this was that a number of staff expressed that the skills they had learnt from the SPI project made them more marketable and that they valued the contribution of the project to their career development.
- Managers felt that they were in a better position to know what sort of training and development their staff required.
- The definition of some of the software engineering processes and practices was seen to provide a *"road map that everyone understands and can follow"* and also created smoother flowing processes that helped bridge the organisational gaps between the software development and business departments.

There was one particular negative outcome raised by the respondents, namely that many people felt that the SPI project slowed down their work. This was partly because the project was seen to intrude on the time required for "real" project work, and because the introduction of formal processes sometimes had the effect of delaying a project. For example, a requirement introduced by the project change control process was that people submit change requests on forms for consideration by a steering committee. This was mentioned as a new practice that slowed projects down. Management seemed willing to accept an initial decrease in productivity resulting from the SPI project and saw this as the result of the learning curve associated with the new practices. One of the managers did state, however, that the learning curve was longer than he had expected.

Finally, in response to the question "*Would you do this again having learnt what you have about the possible outcomes of an SPI initiative?*" all the respondents answered positively, indicating that the general sentiment was one of SPI being worth the effort, even if concrete benefits in terms of performance improvements had not yet been realised.

Although the jury is still out on whether the SPI project added any real value in terms of increased productivity and quality, it seems that quite a few "soft" benefits were realised by the project before it was terminated. Many of these outcomes fall into the quality of work life category of outcomes or are similar to the intangible benefits described in section 1.2.3. One could perhaps conclude that it is possible to realise these sorts of outcomes in a much shorter timeframe than that in which increased software process maturity levels or improved software process outputs might be realised. These "short-term", intangible benefits could, in turn, increase commitment to the SPI project and help sustain longer-term investments in an on-going SPI initiative.

4.3 The Updated Extended SPI Model (ESPIM)

The discussion on the findings of the case study concludes with suggested changes to the ESPIM, previously depicted in tables 2-9 and 2-10. Since changes were not required to the essential structure of the SPIF, as defined during the literature study, the structure of the ESPIM has not changed either, and the same representation that was used previously is used to represent the updated ESPIM.

The updated model, having taken cognisance of the refinements suggested as a result of the case study, is shown in tables 4-3 and 4-4. Changes to the original ESPIM are shown in bold text and areas for which insufficient evidence was gathered are shown in italic text. In these latter cases the ESPIM was not changed and the conclusions of the literature study are maintained. Normal text implies that the ESPIM was not changed in that area since the conclusions of the literature study could be corroborated by the case study.

In conclusion, the SPIF and the ESPIM were largely found to be valid, with most of the changes suggested by the case study affecting the non-process change dimensions. Here the situation was generally that some of the non-process change dimensions could actually be aligned with a lower maturity level than that suggested by the literature. The number of additions to the ESPIM were fairly limited and typically involved very specific changes. These included an additional pre-requisite process for systematic reuse, i.e. requirements analysis, and the ability of the SEPG to respond quickly to unanticipated changes and opportunities.

The implications of the findings described in this chapter are discussed in chapter 5.

Framework Component	SPI Elements or Considerations
Change Environment	
Strategic Initiatives	<p><i>Driven by a clear business need and aligned with a business strategy.</i></p> <p><i>Measurable objectives are set.</i></p> <p>Top management leadership and sponsorship to SPI.</p> <p>Organisation-wide buy-in and commitment.</p>
Change Resources	<p>SPI costs and benefits have been determined and are understood.</p> <p>Long-term commitment to improvement resources and funds is obtained.</p>
Change Facilitators	<p>A culture and management orientation inclined toward change and innovation.</p> <p>A learning capacity in the organisation.</p> <p>A knowledge sharing capability.</p> <p>Higher levels of learning-related scale, related knowledge and diversity.</p>
Change Context	<p>Elements of an SPI friendly change context include:</p> <ul style="list-style-type: none"> · Sound business imperatives to embark on SPI. · Corporate strategies that support the use of IS for business advantage. · The presence of complementary corporate improvement initiatives. · A culture that facilitates continual improvement and innovation. · The resources and size to carry an SPI initiative. · A dependance on software. · Appropriate IS resources, organisational structures, policies and practices. · IS staff with the required skills and an inclination towards SPI.
Change Implementation	
Software Processes	<p>SW-CMM or SPICE Model-based software process improvement</p> <p>A software process improvement process</p>
Non-Process Change Dimensions	<p>Complementary changes can be made in the following areas:</p> <ul style="list-style-type: none"> · Structure · People · Information and Technology · Management · Systematic Software Reuse · <i>Enterprise Architectures</i> · <i>External Relationships</i>
Change Management	<p>Strategies that overcome resistance to change and then sustain the change.</p> <p>A change management process that addresses anticipated, emergent and opportunity-based types of change and is integrated with the SPI process.</p> <p>Formal change management strategies should be complemented with informal, re-inforcing actions, especially by management.</p>
Change Outcomes	<p><i>Increased process maturity and capability.</i></p> <p><i>Improved process outcomes.</i></p> <p>Improved Quality of Worklife.</p> <p>Other intangible benefits.</p>

Table 4-3: The Updated Extended SPI Model

CMM Key Process Areas			SPICE Process Attributes		Non-Process Change Dimensions			
Maturity or Capability Levels			Structure	People	Information and Technology	Management	Other	
1 Ad hoc or Performed informally	<ul style="list-style-type: none">None	<ul style="list-style-type: none">Process Performance	<ul style="list-style-type: none">SDWTsManagement Steering Committee for SPIProcess OwnersProcess Improvement TeamsSoftware Engineering Process Group	<ul style="list-style-type: none">Baldwin et al's (1997) Stage I learning strategiesProvide training in SPI and qualityReward focus on process, SPI and quality	<ul style="list-style-type: none">Software Process Management tools	<ul style="list-style-type: none">SPI friendly management orientation, strategy, structures, systems and practicesUse of performance loop information	<ul style="list-style-type: none">Exploitation of external alliances with more mature organisations	
2 Repeatable or Managed	<ul style="list-style-type: none">Requirements ManagementSoftware Project PlanningSoftware Project Tracking & OversightSoftware Subcontract ManagementSoftware Quality AssuranceSoftware Configuration Management	<ul style="list-style-type: none">Performance ManagementWork Product Management	<ul style="list-style-type: none">Centres of excellenceProject office	<ul style="list-style-type: none">Reward meeting project schedule & quality plansCompetency-based workforce planning and recruitment for level 2 processesCompetency-based training for level 2 processes	<ul style="list-style-type: none">Automated tools that support SW-CMM level 2 KPAs or the SPICE Management and Support Process Categories			

Maturity or Capability Levels	CMM Key Process Areas	SPICE Process Attributes	Non-Process Change Dimensions				
			Structure	People	Information and Technology	Management	Other
3 Defined or Established	<ul style="list-style-type: none"> • Organisation Process Focus • Organisation Process Definition • Training Programme • Integrated Software Management • Software Product Engineering • Intergroup Co-ordination • Peer Reviews 	<ul style="list-style-type: none"> • Process Definition • Process Resource 	<ul style="list-style-type: none"> • <i>Team Building KPA (P-CMM)</i> • <i>Team-Based Practices KPA (P-CMM)</i> 	<ul style="list-style-type: none"> • <i>Competency-based workforce planning and recruitment</i> • <i>Competency-based training</i> • <i>Baldwin et al's (1997) Stage II learning strategies</i> • <i>Reward use of standard processes</i> 	<ul style="list-style-type: none"> • CASE and I-CASE tools and IPSEs to support all defined processes 	<ul style="list-style-type: none"> • <i>Use of relevance loop information</i> • <i>Use of benchmarks</i> 	<ul style="list-style-type: none"> • Systematic software reuse • <i>Use of Enterprise Architectures</i>
4 Quantitatively Managed or Predictable	<ul style="list-style-type: none"> • Quantitative Process Management • Software Quality Management 	<ul style="list-style-type: none"> • Process Measurement • Process Control 		<ul style="list-style-type: none"> • <i>Reward meeting process & quality targets</i> 	<ul style="list-style-type: none"> • <i>Process Simulation tools</i> 		
5 Optimising	<ul style="list-style-type: none"> • Defect Prevention • Technology Change Management • Process Change Management 	<ul style="list-style-type: none"> • Process Change • Continuous Improvement 		<ul style="list-style-type: none"> • <i>Baldwin et al's (1997) Stage III learning strategies</i> • <i>Reward proactive innovation and improvement</i> 			

Table 4-4: Updated mappings from the software process maturity or capability levels to the non process change dimensions

5. Conclusion

The research conducted during this dissertation explored the assumption that the SW-CMM and SPICE SPI models appear to have a very narrow focus on the software process itself, and do not seem to address issues pertaining to the environment within which SPI occurs, or other areas that could contribute to improved software development and maintenance performance. In exploring this perceived gap a theoretical framework for extending the SW-CMM and SPICE models was developed to explicitly consider change aspects other than those pertaining directly to the software process itself. The resultant SPI Framework (SPIF) was strongly influenced by the software process, business process change and IS organisation transformation literature, and consists of three main components: the *change environment*, the *change implementation* and the *change outcomes* components. The SPIF was depicted in figure 1-2.

During the course of the literature study the an Extended SPI Model (ESPIM), based on the structure of the SPIF, was created. The ESPIM addresses complementary improvement areas that the SW-CMM and SPICE SPI models do not address in detail. The ESPIM also shows how the improvement areas could be related to these models and be aligned with the different maturity levels of these models. The ESPIM, depicted in Tables 2-8 and 2-9, reflects the conclusions of the literature study.

The ESPIM was used as the basis for a case study that was conducted with the purpose of testing and refining the SPIF and ESPIM. The findings of this study, described in chapter 4, culminated in an updated version of the ESPIM. This chapter discusses the implications and limitations of the research, identifies a number of potential areas for future research and ends with a few concluding remarks.

5.1 Implications and Limitations of the Research

5.1.1 Implications for Theory and Research

The conclusions and implications of the research can, at best, be regarded as preliminary, since they are only based on one case study and require further examination with a wider sample of SPI projects in different contexts. Nonetheless, the case study raised a few interesting points. In particular, the following aspects are worthy of mention.

The overall structure of the ESPIM was borne out by the case study in that meaningful observations were made for each area of the model and none of the areas were shown to be inappropriate or irrelevant to the execution of an SPI initiative or its outcomes. The ESPIM was also shown to provide a good basis for conducting a case study in the practice of SPI and the impact of supporting change dimensions. Although more empirical work is required to elaborate and verify the ESPIM, it is believed that a useful starting point has been made in the development of a model that can be used to guide research in this area.

There were no significant deviations or new findings in the *Change Environment* parts of the ESPIM. What was clear, however, was that some items were found to impact the execution or overall effectiveness of the SPI project very strongly. For, example, the presence of formal sponsorship was identified as having a significant impact on the ability of the SPI project to function effectively. Further research is required to identify those elements in the change environment that have the largest impact on an organisation's ability to execute an SPI project and the outcomes of such a project successfully. This will help identify critical enablers and inhibitors to an SPI project.

The ESPIM was challenged in a few situations. In the *Change Implementation* area of the model, variations highlighted by the case study tended to fall into one of two categories: changes to the alignment between items and additions to the model.

- *Changes to the alignment between items* Here the alignment between items, at different maturity levels, in the software process and non-process change dimensions in the case study were found to be different from the alignment predicted by the literature study conclusions. For example, the presence of an SEPG, described as an item in the structure non-process change dimension, was realigned with level 1 from level 3 during the case study.

Further exploration on the alignment between different processes and practices at each maturity level, and the impact of these alignments, is required, especially in situations other than the one studied here. Also, this research emphasised the alignment between only the software process and non-process change dimensions. The validity of the implied alignment between items in the non-process change dimensions should also be explored.

- *Additions to the model* Two possible types of addition to the ESPIM came to light during the case study: new items, not previously identified, and new insights and detail to existing items. The suggestion that requirements analysis is also an enabling process of Systematic Reuse is an example of a case where additional insights came to light. The suggestion that using a project office at level 1 is very helpful in moving to level 2 is an example of a new item in the ESPIM. Researchers should seek opportunities to fill out the ESPIM over time, both with new items and with additional detail to all the items.

A few issues were raised in the *change outcomes* area of the ESPIM. Firstly, the types of outcomes and benefits that one may expect were not necessarily comprehensively identified, for example, the case study identified a number of "soft" and intangible benefits not previously mentioned. Secondly, the order in which one can expect the benefits to accrue had not been explored. For example, quality of work life and other intangible benefits seem to be realised before increased process maturity and process outcomes. This implies the need for further research into the nature of the possible change outcomes, the actual outcomes, and the order in which the outcomes may be realised.

The research has shown that the SPIF and the ESPIM provide a valuable starting point for further research into the development of the SW-CMM and SPICE models and the execution of SPI initiatives. The following are some of the ways that the framework and model can be used:

- To *guide research* into the theory and practice of SPI and IS organisation performance improvement and provide a unifying theme and context for this research. For example, it is possible that the SPIF can be used as the guide a number of studies into SPI, just as Kettinger and Grover's (1995) Framework of Business Process Change Management has been used as the basis for a number of studies. For example, Janz et al (1997) used Kettinger and Grover's (1995) framework to investigate the link between autonomous teams and business process outcomes, and Guha et al (1997) used it in the context of research into the various facilitators and inhibitors to the success of business process change exercises.

- To *integrate a number of research frameworks and models* from related fields. In this way, a multitude of different maturity models and stages of growth-type models (for example those described by Galliers and Sutherland (1994) which addressed many of the elements contained in the ESPIM) could be investigated to identify possible relationships and dependencies between their elements and/or stages in the context of IS organisation renewal and SPI. For example, the ESPIM could be used in the context of research into ways in which the SW-CMM or SPICE and the People Capability Maturity Model (P-CMM) could be used in conjunction with each other. Such integrated models will offer a very powerful and comprehensive view of how to improve IS organisations.
- To *describe* the various elements of an approach to SPI, and to a lesser extent the interrelationships between them. It thereby provides insights for both researchers and practitioners into the various elements and considerations of an SPI initiative.

The SPIF and ESPIM can therefore be seen as a *research*, an *integrating* and a *descriptive* framework and model. Although the conclusions of this research can only be seen as initial findings that must be subjected to further investigation, the SPIF and ESPIM seem to stand up to the test of the case study, to provide a valuable research tool, and to point towards a number of opportunities for further research work.

5.1.2 Implications for Practitioners

The research described in this dissertation is practically applicable. Not only does it challenge practitioners to extend their SPI initiatives to consider a range of related concerns, but it offers them a descriptive model of what these concerns may be and when they should be relevant.

The benefits of considering the results of this research during the practice of SPI include the following:

- The importance of change environment issues is highlighted. The findings in the change environment sections of the ESPIM give practitioners guidance as to which aspects of their environment they need to change or emphasise in order to increase their chances of success during an SPI initiative. It may also imply that practitioners have to concentrate on "getting aspects of the environment right" before embarking on a wholesale improvement initiative, or ensure that action is taken to help overcome the limitations implied by the environment. In other words, the findings of this research could be used by practitioners to assess their environmental readiness for change, to identify critical environmental factors to their change initiatives, and to ensure that the environmental factors are adequately addressed.
- Complementary improvement areas that should be considered in addition to the software process are highlighted. This will prevent a narrow focus on the software process and will help practitioners consider other improvement areas or changes that should be considered during an SPI initiative. In this way practitioners may adopt a more systemic view of SPI and IS organisation improvement.

- Conversely, a means of co-ordinating and integrating parallel improvement exercises is provided. It is not uncommon, especially in large IS organisations, for a number of different improvement exercises, in areas described by the ESPIM, to be taking place at the same time. All too often no attempt is made to align these efforts to ensure that they are mutually re-inforcing and that possible synergy between them are exploited. The SPIF and ESPIM can help to highlight the opportunities to co-ordinate and align these initiatives, and to provide guidance on how this could be done. In this way the research findings to help practitioners reduce duplication, ensure that all initiatives are supportive of each other and do not undermine each other, and establish a cohesive change plan for the IS organisation.
- In a similar vein, important considerations for the introduction of a whole range of possible changes in the IS organisation are offered. For example, when considering the introduction of a CASE tool, the ESPIM would suggest that mature software processes and methodologies are important facilitators of success.

The bottom line value of the research may be that they help "software process bigots", such as the researcher, to realise that significant organisational change is a complex endeavour which may require a broad-based approach to improvement, even though SPI and increased process maturity may be critical, if not central, aspects of this sort of change.

Finally, the value of the SPIF and ESPIM is not necessarily to provide practitioners with a deterministic model of SPI, especially given the relative immaturity of the research into SPI and the variations in the situations of different organisations, but may be to provide managers and practitioners with insight into the situational conditions and improvement "levers" that will enable an SPI initiative to succeed.

5.1.3 Limitations

The findings and conclusions of this research should be, at best, regarded as preliminary, and are subject to a number of limitations. These limitations, together with some suggested additional research that could help overcome them, are discussed below.

- *Only one case study was performed.* Performing a single case study made it possible to cover the entire breadth of the SPIF within the confines of this dissertation. However, basing firm conclusions on a single case study, especially when environmental factors can have such a large impact on the results, is always going to be risky and subject to criticism.

Ideally a number of case studies based on the SPIF and ESPIM will be performed over time in order to test and refine the framework and model further. Also, the availability of multiple case studies will enable cross-case analysis, which can be used to detect similarities and compare differences between SPI initiatives with the purpose of developing a greater understanding of the factors that appear to be widely applicable and those that are entirely context dependent.

- *The case study was not based on a formal SW-CMM or SPICE-based SPI initiative.*

Although the SPI project studied was heavily influenced by both the SW-CMM and SPICE models, it did not explicitly follow them in all ways. The case study has therefore not investigated a "pure" SW-CMM or SPICE-based SPI initiative.

Since the ESPIM is largely based on the SW-CMM or SPICE models, future case studies should also seek to use SPI initiatives that are formally guided by these models in order to further test the validity of the ESPIM.

- *The case study did not span an entire SPI initiative.* The time period covered by the case study was relatively short in SPI terms and did not even span the period typically required to mature from one level to the next, never mind to mature through all the software process maturity levels. This implies that the whole ESPIM was not tested in its entirety, especially at higher software process maturity levels.

To overcome this limitation it is recommended that longitudinal studies of SPI initiatives that span the growth of an IS organisation through a number of software process maturity levels be done. Ideally, this would imply that data be collected as organisations move through each maturity level. This would be useful because it would test the ESPIM at all maturity levels and would reveal patterns of change in the various areas of the SPIF over time.

Unfortunately, there are a number of difficulties inherent in performing such a longitudinal study, not least of which are the high cost and the fact that it requires a much longer period than is typically available to graduate students (Jurison, 1996). A possible way of overcoming this is to integrate the work of a number of researchers in this area using the SPIF and ESPIM to guide the effort

- *The research did not necessarily test all the elements of the Proposed SPI Framework in-depth.* The research emphasised covering the breadth of the SPIF and ESPIM and did not necessarily delve into any of the elements in great detail. Also, very little attention was paid to the interrelationships between the elements in the framework and model the impacts that they may have on each other.

This suggests a number of potential areas of future research. Any one of the elements in framework or model, or the relationships between the elements, could be subjected to in-depth investigation in the context of SPI.

In summary, many of the limitations of this research were a result of the massive area covered by the dissertation. This field of study offers many opportunities for research that could keep a comprehensive research programme occupied for an extended period. The scope of the dissertation, and the fact that one of its implied goals was to be exploratory in nature and cover a large swathe of the research territory, therefore caused a number of the limitations discussed above. Nonetheless, the results of the research appear promising and have potentially uncovered fertile ground for future research in this field.

5.2 Concluding Remarks

This dissertation set out to explore the following core research questions.

- *What should the change environment for SPI look like?*
- *What other change dimensions should be considered in conjunction with the software process to facilitate SPI?*
- *How should these other change dimensions be aligned with the software process maturity levels described by the SW-CMM and SPICE models in the sense of being complementary and supportive of the goals of each maturity level?*

Table 5-1 summarises the research objectives which were defined to assist in answering these questions. It also shows the ways in which the dissertation has attempted to satisfy them.

Research Objective	Satisfied by:
Identify complementary improvement considerations that the SW-CMM and SPICE SPI models do not necessarily address in detail.	The development of the Proposed SPI Framework and the Extended SPI Model during the literature study (Chapter 2)
Relate the identified improvement considerations to the SPI models.	
Populate the Proposed SPI Framework with the improvement areas.	
Conduct a case study to test and, potentially, refine the Extended SPI Model.	The case study and analysis of the findings (Chapter 4)
Identify opportunities for future research.	The suggestion of a number of areas for future research in the discussion on the implications and limitations of the research. (Chapter 5)

Table 5-1: Research objectives

The development and refinement of the SPIF and ESPIM during this research have contributed to answering the core research questions listed above in that they addressed the change environment for SPI, identified improvement dimensions that complement SPI, and attempted to align these improvement dimensions and the SPI maturity levels. However, the issues raised by the questions have not necessarily been completely explored and, at the same time, the research has raised a number of additional questions. This presents researchers who have an interest in this field with many potentially interesting lines of enquiry in the future.

In conclusion, increasing software process maturity is on the agenda of many software organisations. This is not a trivial exercise, and potentially impacts many facets of the organisation. The value of this research is that it recognises this fact and strives to provide researchers and practitioners with a view of SPI that sees the practice of SPI as a comprehensive programme of organisational change. In this way it describes the SPI playing field holistically, offers insights into the potential relationships between the various components of such a change programme, and, in the process, establishes a foundation for future growth in SPI research and practice.

References

- Adler, P. S., McDonald, D. W. and MacDonald, F., Strategic Management of Technical Functions, *Sloan Management Review*, Winter 1992
- Ahlgren, M., *Reaching CMM Level 3 - the Ericsson Telecommunicatie B.V. Case*, <<http://www.espi.co.uk/nonmembers/spider/papers/eric2.htm>>, (Accessed 14 November 1997)
- AlBanna, S. J. and Osterhaus, J., Meeting the Software Challenge: A Model for IS Transformation, *Information Systems Management*, Winter 1998
- Anonymous, *Comparison Between Different Maturity Models and Methods*, <<http://www.espi.co.uk/nonmembers/spider/papers/qlabs3.htm>>, (Accessed 14 November 1997)
- Anonymous, *SPICE: An Overview to the Document Suite*, <http://www-sqi.cit.gu.edu.au/spice/suite_view.shtml?>, (Accessed 20 November 1997), 1997b
- AppleGate, P. D. and Zawacki, R. A., A Balanced Approach to IT Transformation, *Information Systems Management*, Summer 1997
- Baldwin, T.T., Danielson, C. and Wiggernhorn, W., The Evolution of Learning Strategies in Organisations: From Employee Development to Business Redefinition, *Academy of Management Executive*, Vol. 11, No. 4, 1997
- Bartlett, C. A. and Goshall, S., Changing Role of Top Management: Beyond Strategy to Purpose, *Harvard Business Review*, Nov/Dec 1994
- Bartlett, C. A. and Goshall, S., Changing Role of Top Management: Beyond Systems to People, *Harvard Business Review*, May/June 1995
- Basili, V. and Green, S., Software Process Evolution at the SEL, *IEEE Software*, July 1994
- Boggis, P and Yarrington, P, Breaking Up is Hard To Do, *Information Strategy*, September 1997
- Brancheau, J. C., Janz, B. D and Wetherbe, J. C., Key Issues in Information Systems Management: 1994-95 SIM Delphi Results, *MIS Quarterly*, June 1996
- Brodman, J. G. and Johnson, D. L., What Small Business and Small Organisations Say About the CMM: Experience Report, *Proceedings of the 16th International Conference on Software Engineering*, 1994
- Brodman, J. G. and Johnson, D. L., Return on Investment from Software Process Improvement as Measured by US Industry, *CrossTalk*, April 1996
- Budlong, F. C., Szulewski, P. A. and Ganska, R. J., *Process Tailoring for Software Project Plans*, <<http://www.stsc.hill.af.mil/ProcessPlan/prplp.html>>, January 1996, (Accessed 4 April 1997)

Burke, S., *Radical Improvements Require Radical Actions: Simulating a High-Maturity Software Organisation*, Technical Report CMU/SEI-96-TR-024, Software Engineering Institute, Carnegie Mellon University, June 1997

Cecere, M., Architecting Architecture, *CIO Magazine*, 15 April 1998

Choi, T. Y. and Behling, O. C., Top Managers and TQM Success: One More Look After All These Years, *Academy of Management Executive*, Vol. 11, No. 1, 1997

Clarke, C. E., Cavanaugh, N. C., Brown, C. V. and Sambamurthy, V., Building Change-Readiness Capabilities in the IS Organisation: Insights from the Bell Atlantic Experience, *MIS Quarterly*, December 1997

Cross, J., Earl, M. J. and Sampler, J. L., Transformation of the IT Function at British Petroleum, *MIS Quarterly*, December 1997

CSC, Future Roles and Responsibilities for the IS Department, *CSC Foundation Final Report 96*, CSC Research and Advisory Services, 1994

Curtis, B., Hefley, W.E. and Miller, S., *Overview of the People Capability Maturity Model*, Carnegie Mellon University, Software Engineering Institute, CMU/SEI-95-MM-01, September 1995

Davenport, T. H. and Beers, M. C., Managing Information About Processes, *Journal of Management Information Systems*, Vol. 12, No. 1, Summer 1995

Davenport, T. H., Jarvenpaa, S. L. and Beers, M. C., Improving Knowledge Work Processes, *Sloan Management Review*, Summer 1996

Deephouse, C., Muhopadhyay, T., Goldenson, D. R. and Kellner, M. I., Software Processes and Project Performance, *Journal of Management Information Systems*, Vol. 12, No. 3, 1996

Diaz, M and Sligo, J., How Software Process Improvement Helped Motorola, *IEEE Software*, September/October 1997

Dion, R., Process Improvement and the Corporate Balance Sheet, *IEEE Software*, July 1993

Downes, V., George, G. W. and Chappell, C., *Best Practice in Software Development: Experiences in European Industry*, Ovum Reports, 1996

Fichman, R. G. and Kemerer, C. F., The Assimilations of Software Process Innovations: An Organisational Learning Perspective, *Management Science*, Vol. 43, No. 10, October 1997

Frakes, W. B. and Fox, C. J., Sixteen Questions About Reuse, *Communications of the ACM*, Vol. 38, No. 6, June 1995

Galbraith, J. R., *Designing Organisations: An Executive Briefing on Strategy, Structure and Process*, Jossey-Bass Publishers, San Francisco, 1995

Galliers, R. D. and Sutherland, A. R., Information Systems Management and Strategy Formulation: The 'Stages of Growth' Model Revisited, *Journal of Information Systems*, Vol. 1, 1991

Goshall, S and Bartlett, C. A., Changing Role of Top Management: Beyond Structure to Processes, *Harvard Business Review*, Jan/Feb 1995

Gremba, G. and Myers, C., *The IDEAL Model: A Practical Guide for Improvement*, <<http://www.sei.cmu.edu/ideal/ideal.bridge.html>>, 1997, (Accessed 30 May 1998)

Guha, S, Grover, V, Kettinger, W. J. and Teng, J. T. C., Business Process Change and Organisational Performance: Exploring an Antecedant Model, *Journal of Management Information Systems*, Vol. 14, No. 1, Summer 1997

Guinan, P. J., Coopridier, J. G. and Sawyer, S., The Effective Use and Automated Application of Development Tools, *IBM Systems Journal*, Vol. 36, No. 1, 1997

Haley, T. J., Software Process Improvement at Raytheon, *IEEE Software*, November 1996

Hefley, W. E., *Where does Team Building Fit as a Component of Mature Software Development Processes?*, <<http://hsb.baylor.edu/ramsower/ais.ac.96/papers/hefley2.htm>>, 1996, (Accessed 16 February 1998)

Henry, E. and Benoit, F., Large-Scale Industrial Reuse to Reduce Cost and Cycle Time, *IEEE Software*, September 1995

Herbsleb, J. D. and Goldenson, D. R., A Systematic Survey of CMM Experience and Results, *Proceedings of the 18th International Conference on Software Engineering*, 1996

Harkness, W. L., Kettinger, W. J. and Segars, A. H., Sustaining Process Improvement and Innovation In the Information Services Function: Lessons Learned at the Bose Corporation, *MIS Quarterly*, Vol. 20, No. 3, September 1996

Humphrey, W. S., *Managing the Software Process*, Addison Wesley, 1989

Humphrey, W. S., *A Discipline for Software Engineering*, , Addison-Wesley, 1995

Humphrey, W. S., Three Dimensions of Process Improvement Part 1: Process Maturity, *CrossTalk*, February 1998

Humphrey, W. S., Three Dimensions of Process Improvement Part 2: The Personal Process, *CrossTalk*, March 1998

Humphrey, W. S., Three Dimensions of Process Improvement Part 3: The Team Process, *CrossTalk*, April 1998

Hunter, R., *SEI's CMM: Forget the Score. Are You in the Right Game?*, Research Note TU-480-1200, Application Development and Management Strategies (ADM), Gartner Group, 9 August 1995

Hunter, R., *Once is Not Enough: Requirements for a Sustainable Reuse Program*, Strategic Analysis Report R-480-129, Application Development and Management Strategies (ADM), Gartner Group, 30 January 1996

Hunter, R., *Year 2000: After the Deluge, Regulated Development?*, Research Note SPA-250-1050, Application Development and Management Strategies (ADM), Gartner Group, 30 July 1997

Hutchings, T., Hyde, M.G., Marca, D. and Cohen, L., Process Improvement that Lasts: An Integrated Training and Consulting Method, *Communications of the ACM*, Vol. 36, No. 10, October 1993

ISO, *Software Process Assessment Part 1: Concepts and Introductory Guide*, Version 2.0, ISO/IEC15504-1, 1996a

ISO, *Software Process Assessment Part 2: A Reference Model for Processes and Process Capability*, Version 2.0, ISO/IEC15504-2, 1996b

ISO, *Software Process Assessment Part 3: Performing an Assessment*, Version 2.0, ISO/IEC15504-3, 1996c

ISO, *Software Process Assessment Part 4: Guide to Performing Assessments*, Version 2.0, ISO/IEC15504-4, 1996d

ISO, *Software Process Assessment Part 5: An Assessment Model and Indicator Guidance*, Version 2.0, ISO/IEC15504-5, 1996e

ISO, *Software Process Assessment Part 6: Guide to Qualification of Assessors*, Version 2.0, ISO/IEC15504-6, 1996f

ISO, *Software Process Assessment Part 7: Guide for Use in Process Improvement*, Version 2.0, ISO/IEC15504-7, 1996g

ISO, *Software Process Assessment Part 8: Guide for Use in Determining Supplier Process Capability*, Version 2.0, ISO/IEC15504-8, 1996h

ISO, *Software Process Assessment Part 9: Vocabulary*, Version 2.0, ISO/IEC15504-9, 1996i

Janzon, T., Nilson, B., Gumpert, J. and Sved, L., *Successful Process Improvement*, <<http://hsb.baylor.edu/ramsower/ais.ac.97/papers/janzon.htm>>, 1997, (Accessed 1 May 1998)

Janz, B. D., Wetherbe, J. C., Davis, G. B. and Noe, R. A., Reengineering the Systems Development Process: The Link between Autonomous Teams and Business Process Outcomes, *Journal of Management Information Systems*, Vol. 14, No. 1, 1997

Jones, C., The Economics of Software Process Improvement, *Computer*, January 1996

Juran, J. M., *Juran on Planning for Quality*, The Free Press, 1988

Jurison, J., The Temporal Nature of IS Benefits: A Longitudinal Study, *Information and Management*, Vol. 30, 1996, pp. 75-79

Kaplan, R. S. and Norton, D. P., The Balanced Scorecard - Measures that Drive Performance, *Harvard Business Review*, January/February 1992

Kettinger, W. J. and Grover, V., Special Section: Toward a Theory of Business Process Change Management, *Journal of Management Information Systems*, Vol. 12, No. 1, 1995

Kotter, J. P., Leading Change: Why Transformation Efforts Fail, *Harvard Business Review*, March/April 1995

- Larner, C. and Bray, K., *Establishing a Continuous Improvement Culture - the Practical Experience*, <<http://www.espi.co.uk/nonmembers/spider/papers/lloyds1.htm>>, (Accessed 14 November 1997)
- Lawlis, P. K., Flowe, R. M. and Thordahl, J. B., A Correlational Study of the CMM and Software Development Performance, *CrossTalk*, September 1995
- Lim, W. C., Effects of Reuse on Quality, Productivity and Economics, *IEEE Software*, September 1994
- Lowe, D. E. and Cox, G. M., Implementing the Capability Maturity Model for Software Development, *Hewlett-Packard Journal*, August 1996
- Magrassi, P., *Architecture: A Foundation for Durable Applications*, Strategic Analysis Report R-560-124, Application Development and Management Strategies (ADM), Gartner Group, 19 April 1995
- Mashiko, Y. and Basili, V. R., Using the GQM Paradigm to Investigate Influential Factors for Software Process Improvement, *Journal of Systems Software*, Vol. 36, 1997
- McGibbon, T., *A Business Case for Software Process Improvement*, <<http://www.dacs.com/techs/roi.soar/soar.html>>, 1996, (Accessed 15 April 1997)
- McGuire, E. G. and Gibson, R., *Research Issues in Software Process Improvement: Qualitative and Quantitative Data Gathering in Empirical Studies*, <<http://hsb.baylor.edu/ramsower/ais.ac.96/papers/mcguire2.htm>>, 1996, (Accessed 7 November 1997)
- Myers, C. R., The Human Element in Technology Adoption, *CrossTalk*, December 1997
- Nerur, S. and Raghupathi, W., *Software Process Improvement: Toward a Comprehensive Framework for Research*, <<http://hsb.baylor.edu/ramsower/ais/papers/nerur.htm>>, 1996, (Accessed 7 November 1997)
- NRaD Software Process Improvement Charter, <<http://sepo.osc.mil/SPIcharter.html>>, 30 January 1996, (Accessed 22 October 1997)
- O'Reilly, C., Corporations, Culture, and Commitment: Motivation and Social Control in Organisations, *California Management Review*, Vol. 31, No. 4, Summer 1989
- Orlikowski, W. J., CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development, *MIS Quarterly*, Vol. 17, No. 3, September 1993
- Orlikowski, W. J. and Hofman, J. D., An Improvisational Model for Change Management: The Case of Groupware Technologies, *Sloan Management Review*, Winter 1997
- Orville, D. D., Process Action Teams: From "Black Holes" to "Shining Stars", *CrossTalk*, February 1998
- Pajerski, R. and Waligora, S., *The Improvement Cycle: Analyzing Our Experience*, <<http://fdd.gsfc.nasa.gov/selres.html>>, (Accessed 7 November 1997)

Paulk, M. C., The Evolution of the SEI's Capability Maturity Model for Software, *Software Process: Improvement and Practice*, Vol. 1, Pilot Issue, August 1995

Paulk, M. C., *Mappings Between ISO 12207, ISO 15504 (SPICE), Software CMM v1.1, and Software CMM v2 Draft C*, Working Paper, 23 February 1998, <<http://www.sei.cmu.edu/activities/cmm/cmm.articles.html>>, (Accessed 23 May 1998)

Paulk, M. C., Garcia, S. M. and Chrissis, An Architecture for CMM Version 2?, *Proceedings of the 8th Software Process Group Conference*, Atlantic City, NJ, 20-23 May 1996

Paulk, M.C., Konrad, M. D. and Garcia, S. M., CMM Versus SPICE Architectures, *Software Process Newsletter*, IEEE Computer Society Technical Council on Software Engineering, No. 3, Spring 1995

Perry, D. E., Staudenmayer, N. A. and Votta, L. G., People, Organisations, and Process Improvement, *IEEE Software*, July 1994

Pfleeger, S. L. and Rombach, H. D., Measurement-Based Process Improvement, *IEEE Software*, July 1994

Process Management White Paper, <<http://www.lbms.com/pefact/whiteppr/toc.htm>>, (Accessed 12 April 1997)

Raffo, D. M., *The Role of Process Improvement in Attaining Strategic Goals and Providing Financial Value*, <<http://hsb.baylor.edu/ramsower/ais.ac.97/papers/raffo.htm>>, 1997, (Accessed 1 May 1998)

Ragan, T. and Reifer, D. J., Adding Product Lines, Architectures and Software Reuse to the Software Acquisition Capability Maturity Model, *Crosstalk*, May 1998

Rahardja, C. P., *A Comparative Evaluation of the Capability Maturity Model and ISO9000*, <<http://erau.db.erau.edu/~rahardjic/grp/ch0.html>>, April 1996, (Accessed 15 April 1997)

Randall, K. A. and McGuire, E. G., *Success Factors and Implementation Strategies for Process Improvement Teams*, <<http://hsb.baylor.edu/ramsower/ais.ac.97/papers/randall.htm>>, 1997, (Accessed 1 May 1998)

Risks in Software Process Improvement Programmes, <<http://www.espi.co.uk/nonmembers/spider/papers/qlabs4.htm>>, (Accessed 14 November 1997)

Robson, W., *Strategic Management and Information Systems: An Integrated Approach*, Second Edition, Pitman Publishing, 1997

Rockart, J. F., Earl, M. J. and Ross, J. W., Eight Imperatives for the New IT Organisation, *Sloan Management Review*, Vol. 38, No 1, 1996

Ross, J. W., Beath, C. M and Goodhue, D. L., Develop Long-Term Competitiveness Through IT Assets, *Sloan Management Review*, Vol. 38, No 1, 1996

Rout, T. P., SPICE: A Framework for Software Process Assessment, *Software Process: Improvement and Practice*, Vol. 1, Pilot Issue, August 1995

- Rubin, H. A., Johnson, M. and Yourdon, E., Software Process Flight Simulation: Dynamic Modeling Tools and Metrics, *Information Systems Management*, Summer 1995
- Saiedian, H. and Kuzara, R., SEI Capability Maturity Model's Impact on Contractors, *Computer*, January 1995
- Sarker, S., *Understanding and Managing I/S Implementation in "Socially Constructed" Organisations*, <<http://hsb.baylor.edu/ramsower/acis/papers/sarkar.htm>>, 1995, (Accessed 1 May 1998)
- SEI, Carnegie Mellon University, Software Engineering Institute, *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, 1995
- SEI, *Process Maturity Profile of the Software Community: 1996 Year End Update*, Carnegie Mellon University, Software Engineering Institute, May 1997
- Shea, K., *Integration of Software Process Maturity and Domain Engineering to Facilitate Reuse within the Enterprise*, <<http://nplace/wvhtf.org/reuse97/KShea/>>, (Accessed 25 November 1997)
- Short, K., *Component Based Development and Object Modeling*, Texas Instruments Software, <<http://www.ti.com>>, February 1997, (Accessed 19 June 1997)
- Stevenson, D. A., *Business Themes and Enterprise Architectures*, Unpublished Master of Commerce Paper, University of Cape Town, 1995
- Stoddard, D. B. and Jarvenpaa, S. L., Business Process Redesign: Tactics for Managing Radical Change, *Journal of Management Information Systems*, Vol. 12, No. 1, 1995
- SW-CMM v2 Draft C*, <<http://www.sei.cmu.edu/activities/cmm/draft-c/c.html>>, (Accessed 10 July 1998)
- Teng, J. T., Fiedler, K. D. and Grover, V., *The Organisational Context of Process Reengineering Project Initiatives*, <<http://hsb.baylor.edu/ramsower/ais.ac.97/papers/teng.htm>>, 1997, (Accessed 1 May 1998)
- Uniframe*, <<http://www.esi.es/Projects/PI/Uniframe/description.html>>, (Accessed 21 March 1998)
- Walker, A. J. and Knight, J., *Software Quality Management: Software Process Improvement and Capability Determination*, Software Engineering Applications Laboratory, August 1997
- Wigle, G. B. and Yamamura, G., Practices of an SEI CMM Level 5 SEPG, *CrossTalk*, November 1997
- Yamamura, G. and Wigle, G. B., SEI CMM Level 5: For the Right Reasons, *CrossTalk*, August 1997
- Zultner, R. E., TQM for Technical Teams, *Communications of the ACM*, Vol. 36, No 10, October 1993

Appendix A: Interview Schedule

The questions used during the case study, grouped by the different components of the Proposed SPI Framework, are listed below. These questions represent the final set of questions after they were reworked to include the lessons learnt during the initial self-interview by the researcher (see the Methodology Procedures in Chapter 3).

The questions assume that the respondents will be briefed about the scope of the case study and that the interviewer will be on hand to rephrase questions or clarify terms where needed. Wherever possible respondents should also be requested to give examples to illustrate and clarify their answers.

1 Preamble

- 1.1 Briefly describe your company, its business and markets, size, environment, etc.
- 1.2 Describe the SPI initiative within this company in terms of which areas it covers (scope), the number of people affected, the type of work that these people do, when it started, who the major driving force behind the initiative was etc.

2 Change Environment

2.1 Strategic Initiative

- 2.1.1 What are the objectives of the SPI programme? Are they measurable and have targets been set? If so, what are they?
- 2.1.2 Is the SPI programme represented in your IS department's strategic plans? If so, in what terms? What are the expectations of this strategy? What business needs motivated the SPI programme?
- 2.1.3 Are the SPI objectives explicitly linked to business objectives? If so, which business objectives? How is this linkage defined and measured?
- 2.1.4 Who sponsors the SPI initiative?
 - Why were they chosen as sponsors?
 - What is their position and responsibilities in the organisation?
 - Are they IT or Business people?

- How would you describe their level of involvement? (active or passive; direction giving or only receivers of status reports; technical or managerial inputs; etc.)
- Do the sponsors actively promote the SPI programme to sub-ordinates and peers? How?

2.1.5 Was the SPI strategy explained and promoted to all affected parties? How? Was this successful in terms of getting their full commitment?

2.2 Change Resources

2.2.1 To what extent has the required cost and time commitment of the SPI programme been determined and communicated to all the stakeholders? In what terms? Is there a budget for the SPI programme? If not, why not? If so, how was it determined? How large is this budget and how many years does it cover?

2.2.2 What people have been committed to the SPI programme?

- A software process engineering group?
- Practitioners allocated to define and implement specific improvements?
- A project manager to manage the SPI programme?
- Management in the capacity of a steering committee?
- Others?

2.2.3 What are the specific responsibilities of the above people in terms of the SPI initiative?

2.2.4 To what extent has their current workload been reduced to enable them to partake in the SPI programme?

2.2.5 To what extent was an initial learning curve and possible slowdown in projects tolerated in the interests of long-term improvements in productivity and time to market?

2.2.6 What other resources have been made available to the SPI programme?

- Extra training (over and above normal levels or specific to SPI)?
- Consultants/contractors to assist in SPI activities or perform software process maturity/capability assessments?
- Other?

2.3 Change Facilitators

2.3.1 Learning Capacity

2.3.1.1 What SPI and software engineering specific training has been provided during the course of the SPI initiative (e.g. SPICE, SW-CMM, software processes, etc.)?

- 2.3.1.2 In what other ways did people learn about SPI and software engineering?
- 2.3.1.3 Have you used pilot projects to experiment with new software processes and practices? How have you incorporated lessons learnt in the further implementation of these processes and practices? Who is responsible for defining and leading this effort?
- 2.3.1.4 Are lessons learnt during the practice of software development and maintenance identified, recorded, communicated and acted on in terms of improving the processes and practices? How is this done? (e.g.'s project completion reports/debriefings, defect analysis, etc.)
- 2.3.1.5 How are software processes identified, defined and customised for your organisation? To what extent are your own staff involved in this, or are "external experts" widely used in this activity?
- 2.3.2 Cultural Readiness
 - 2.3.2.1 Are cultural norms that encourage improvement and innovation in place? Specifically:
 - Is risk taking and failure tolerated? How was this evidenced in the SPI programme?
 - Are key staff members (leaders, managers, "whiz kids") open to change and new ideas? To what extent? If not, why not?
 - Is the vision for a mature software organisation, in SPI terms, shared by all staff members (by both management and staff)? How is the vision promulgated? To what extent is the vision understood by all stakeholders?
 - Does management encourage and enable staff members to make improvements to the software processes and practices?
 - 2.3.2.2 How have the above been demonstrated in the SPI initiative?
- 2.3.3 Knowledge Sharing Capability (and the use of IT to facilitate this)
 - 2.3.3.1 What software process definitions, best practices, lessons learnt and information is shared? How/what mechanisms are used? (e.g. performance measures, processes and practices, lessons learnt, metrics, etc.)
 - 2.3.3.2 Is there a metrics database in place to record process measurements? If so, who uses the processes measures and for what?
 - 2.3.3.3 Is there a library for shared software processes, practices and standards? If so, what form does this library take and who uses the library for what purposes?

- 2.3.3.4 Does your organisation use software process management tools, such as Process Engineer, to document, distribute, execute and track software processes?
- 2.3.3.5 To what extent is technology used to facilitate collaboration and knowledge/information sharing during software development and maintenance or software process improvement exercises? (e.g. use of groupware, e-mail, discussion databases, etc.)
- 2.3.4 Overcoming Knowledge Barriers
- 2.3.4.1 How large is the organisation targeted by the SPI initiative?
- 2.3.4.2 How are the costs accounted for? Who pays for what?
- 2.3.4.3 What other work is happening, or has happened in similar or areas related to SPI (e.g. implementation of methodologies, CASE, project management practices, etc.)? How have these activities assisted or hampered your SPI programme? Why?
- 2.3.4.4 What exposure has your organisation had to software engineering practices and SPI from other quarters (e.g. from consultants, vendor companies, joint-venture companies)? Did this exposure lead to your organisation making changes to its software processes and practices? If so, what changes did you make?
- 2.3.4.5 How diverse is your staff's IS knowledge and experience? (Are all your skills developed in-house? If so, why? Do you explicitly recruit staff from many different institutions and companies? If so, why? What is the educational background of your staff?)
- 2.3.5 Management Orientation
- 2.3.5.1 Who are your main clients, and what do you do for them?
- 2.3.5.2 How do your IS managers see the world in terms of:
- Their view or perception of their clients, and their relationship with these clients,
 - Their main concern regarding their IS department's future role and/or position in the company,
 - Where they tend to apply their minds in terms of strategic issues, operational issues or problems related to past actions and relationships?
 - To what extent does your organisation perform research and development or experimentation to develop new and innovative IT solutions for the business? Give examples of where you have created new business opportunities or significant competitive advantage through this sort of work?
- 2.3.5.3 Give reasons and specific examples to support your answers?

2.3.5.4 Why did you embark on the SPI initiative, and what factors may cause you to increase or decrease your commitment to it?

2.4 Change Context

- 2.4.1 How would you describe the context/environment within which your department operates, and the impact of this on your SPI programme?
- 2.4.2 Are there competitive, customer-driven, or regulatory reasons for improving your IS performance or obtaining a higher software process maturity or capability level? If so, what are they?
- 2.4.3 How important is IT regarded to the success of your organisation? Is it used for competitive advantage? Is software your primary business? Is technology high on executive management's agenda?
- 2.4.4 Are there any significant TQM, productivity improvement or cost cutting initiatives underway in the organisation? Is SPI seen to be part of, or complementary to, this?
- 2.4.5 How is your IS organisation structured? Centres of excellence, project office, support functions, etc.? How do (or do not) the different elements support or facilitate SPI?
- 2.4.6 What IS policies and procedures enforce or support SPI? How?
- 2.4.7 What is the staff's background in SPI and/or TQM (training, education, experience, etc.)? Is there a training plan to develop SPI and software engineering skills?

3 Change Implementation

3.1 Software Processes

- 3.1.1 Are you using a software process improvement model such as the SW-CMM or SPICE to guide your SPI efforts? Whose idea was it? Have you made use of outside expertise in the model that you are using?
- 3.1.2 Why did you choose the specific SPI model that you are using?
- 3.1.3 To what extent are you applying the SPI model rigorously, or are you deviating from it? How?
- 3.1.4 Which software engineering practices have you implemented as part of the SPI initiative (refer to the SW-CMM KPAs or the SPICE processes, or their equivalents)? Have you defined and applied any software processes not explicitly covered by the SW-CMM or SPICE models? Why? Did you realise the expected benefits from this?
- 3.1.5 What is the software process maturity/capability level of your organisation? How was this assessed?

- 3.1.6 What approach, process or methodology are you following to guide your SPI initiative? Which elements of the SEI's IDEAL model or SPICE's SPI steps does it contain?

3.2 Non-Process Change Dimensions

3.2.1 Structures

- 3.2.1.1 Describe the IS and business unit organisational structure and main roles in the area covered by the SPI initiative that were not covered in question 2.4.5.
- 3.2.1.2 Does the organisational structure contain elements of centres of excellence (competency-based organisational units with no direct responsibility for software development or maintenance) and projects (as temporary organisational structures to deliver results) as two separate concepts?
- 3.2.1.3 What is the culture of project teams? Is the project manager a dominant role with all decision making power; or are they run as self-directed teams where, for example, decision making is a team activity and the team is jointly held accountable for the final outcome or their work.
- 3.2.1.4 What structures are in place to support the SPI initiative? Steering committee, SEPG, improvement teams, process owners, other? Who is included in each of these structures and what is the purpose of each one?
- 3.2.1.5 In what ways did the above support, facilitate or hamper your SPI initiative? Why?

3.2.2 People

- 3.2.2.1 To what extent are the organisation's software processes and practices explicitly considered when recruiting staff? Or are recruitment choices more typically based solely on the technologies used? How is a skills match determined?
- 3.2.2.2 Do training plans exist for different roles involved in developing and maintaining software? Are these training plans guided by the software processes and practices of the organisation?
- 3.2.2.3 What training, explicitly targeted at developing an understanding for SPI and quality management, etc. is provided to staff?
- 3.2.2.4 What is the purpose and focus of the training given to the staff involved in all aspects of the development, management and maintenance of software? How much of the training is related to IT issues compared to business issues? What determines the nature of the training provided? Current organisational standards and procedures, future development requirements, etc.? How are these determined?
- 3.2.2.5 What is the level of uncertainty and rate of change in your client's business environment?

3.2.2.6 Are performance appraisals tied to SPI and is the attainment of SPI objectives rewarded in anyway? If so, how? What are some of the specific SPI objectives that are rewarded?

3.2.2.7 In what ways did the above support, facilitate or hamper your SPI initiative? Why?

3.2.3 Information and Technology

3.2.3.1 What software development and maintenance tools do you use?

- CASE
- Integrated development environments
- Test tools
- Defect tracking tools
- Configuration management tools
- Project management tools
- Debuggers
- Requirements management tools
- Other

3.2.3.2 How important are the tools to your software development and maintenance efforts? Have they met all of your initial expectations? If not, why not?

3.2.3.3 Has the effectiveness of the tools changed over time? How and why?

3.2.3.4 Are the organisation's software processes and practices customised to explicitly take advantage of the tools' purpose and features?

3.2.3.5 Are SPI tools used?

- Software process management tools
- Metrics tools
- Software process simulation tools
- Other

How are they applied and what benefits do they deliver? Has their usage changed over time? How and why?

3.2.3.6 In what ways did the above support, facilitate or hamper your SPI initiative? Why?

3.2.4 Management

- 3.2.4.1 What management practices support the SPI initiative? (Sponsorship, leadership, resources, use of rewards, measurements, establishment of SPI policies and procedures, etc.)
- 3.2.4.2 How are measurements used to support SPI? Are process information and measures used to track and improve process performance (performance loop)? Are process information and measures used to tie process goals and outcomes to business goals (relevance loop)? How?
- 3.2.4.3 Do you use software process performance and maturity benchmarks? Why and how? What have the benefits of using the benchmarks been?
- 3.2.4.4 In what ways did the above support, facilitate or hamper your SPI initiative? Why?
- 3.2.5 Reuse
 - 3.2.5.1 Do you apply software reuse in your organisation? What is reused? How widely (within individual projects or across projects over time)? How much reuse is achieved?
 - 3.2.5.2 How is reuse organised and managed in your organisation? How is it motivated?
 - 3.2.5.3 Is there a reuse methodology, or are reuse practices tied into or integrated with the organisation's software processes and practices? How and why?
- 3.2.6 Enterprise Architectures
 - 3.2.6.1 Does your organisation make use of enterprise architectures? How? What "methodologies" do you use to develop, maintain and use your enterprise architectures?
 - 3.2.6.2 What is included (represented) in your enterprise architecture?
 - 3.2.6.3 How are enterprise architecture practices integrated with the organisation's software processes and practices? What does this "buy" you?
- 3.2.7 External Alliances
 - 3.2.7.1 What sort of external alliances pertaining to IT does your organisation use? What is the primary purpose of these alliances?
 - 3.2.7.2 To what extent do you exploit these alliances to improve your own IT capability? How?
- 3.2.8 General
 - 3.2.8.1 What major changes, not directly related to the SPI initiative, but pertaining to the functioning of your IS department were introduced during the time period of the SPI initiative?
 - 3.2.8.2 What motivated these changes?

3.2.8.3 Were they linked to the SPI initiative in any way? How? Why? Who had the responsibility for the co-ordination?

3.3 Change Management

- 3.3.1 Do you have a strategy in place to overcome resistance to the changes brought about by SPI and to sustain these changes?
- 3.3.2 What are the elements of this strategy? (Communication, benefit articulation, widespread practitioner involvement, ensuring benefits, clarity, accessibility & wholeness of change elements, visible sponsorship, training and coaching, rewards, etc.)
- 3.3.3 Did you use a change management process, or were change management activities integrated into the SPI initiative's plans? How? (cf. to the elements in Kotter's (1995) change model).
- 3.3.4 Did you recognise unanticipated changes? How? How did you respond to them?

4 Change Outcomes

- 4.1 What are the expected outcomes from your SPI initiative? (increased software process maturity, improved software process performance, improved QWL)
- 4.2 How were these expected outcomes defined? Why in these terms?
- 4.3 Do you actually know the outcomes of the SPI programme? Why or why not? Where the outcomes measured, or is your perception of the outcomes based on subjective observations?
- 4.4 Were the expected outcomes realised? If not, why not?
- 4.5 Were there any negative outcomes or side effects from the SPI initiative?
- 4.6 Did you realise any unexpected benefits?
- 4.7 How did your clients experience the benefits of your SPI initiative?
- 4.8 How did your staff experience the benefits of your SPI initiative?
- 4.9 Would you do this again having learnt what you have about the possible outcomes of an SPI initiative? What would you change?